HAL open science

Uncertainy 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. Uncertainy quantification for functional dependent random variables. 2014. hal-01075840v1

HAL Id: hal-01075840

https://hal.science/hal-01075840v1

Preprint submitted on 7 Nov 2014 (v1), last revised 28 Jul 2016 (v2)

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.

Uncertainy 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

November 7, 2014

Abstract

This paper proposes a new methodology to quantify the uncertainties associated to multiple dependent functional random variables, linked to a quantity of interest, called the covariate. The proposed methodology is composed of two main steps. First, the functional random variables are decomposed on a functional basis. The decomposition basis is computed by the proposed Simultaneous Partial Least Squares algorithm which enables to decompose simultaneously all the functional variables. Second, the joint probability density function of the coefficients of the decomposition associated to the functional variables is modelled by a Gaussian mixture model. A new method to estimate the parameters of the Gaussian mixture model based on a Lasso penalization algorithm is proposed. This algorithm enables to estimate sparse covariance matrices, in order to reduce the number of model parameters to be estimated. Several criteria are proposed to assess the efficiency of the methodology. Finally, its performance is shown on an analytical example and on a nuclear reliability test case.

1 Introduction

In a large number of fields, like physical or environmental sciences, computer codes prove to be an invaluable tool to model and predict studied phenomena. With the development of computer abilities, numerical simulators have become more and more complex. To describe the characteristics of the studied phenomenon, a great amount of input parameters of various types is needed: scalar, functional, categorical... These characteristics are not perfectly known and the parameters which describe them are thus uncertain. As they can have a great influence on the computer code output, the study of these uncertainties, thanks to uncertainty quantification or sensitivity analysis (De Rocquigny et al. 2008; Saltelli et al. 2000), is an invaluable tool to validate, simplify and better understand a model. The knowledge of the uncertainties associated to each parameter and of their influences on the code output also enables to guide the efforts of characterization of input parameters.

The objective of the work presented in this paper is to quantify and to model the uncertainties associated to functional random variables. The quantification of uncertainties in the context of functional inputs has been studied in some recent works. Anstett-Collin et al. (2013) consider that the functional variables under study are Gaussian processes, and approximate them by a Karhunen-Loève decomposition (Loève 1955). As the variables are Gaussian processes, their coefficients on this functional basis are independent and normally distributed. In their work, this modelling is used to conduct a sensitivity analysis on the output of the computer code that takes the studied functional variables as inputs. Hyndman and Shang (2010) propose a visualization method of functional variables based on their characterization. They decompose the data on the two first components of a functional principal components analysis basis (Ramsay and Silverman 2005). The joint probability density function of the couples of coefficients

is computed thanks to kernel density estimation. In this document, the problem under consideration is different from the previous ones in the sense that the functional random variables to be characterized are dependent and are linked to a scalar (or vectorial) variable, called hereafter a covariate. This covariate can be, for instance, the output of a computer code which takes as inputs the functional random variables. The main objective of this work is thus to provide a new methodology to characterize the uncertainties associated to dependent functional variables linked to a covariate.

The proposed characterization process is composed of two parts. First, the dimension of the problem is reduced by decomposing the functional random variables on a functional basis. In order to take into account the dependence between the functional random variables, the decomposition is done simultaneously on all the variables. This means that the decomposition is done on a vector of functional random variables instead of a unique functional random variable. The link between the functional random variables and the covariate is taken into account in this first step, using specific decomposition. The functional random variables are approximated by their coefficients on the basis. Thus, the problem becomes multivariate instead of multivariate functional. The second step consists in estimating the joint probability density function of the decomposition coefficients. For this, a Gaussian mixture model is proposed and an estimation method based on a penalization algorithm is developed. In addition to providing a characterization of the joint distribution of the studied random variables, this methodology allows also to simulate new realizations of these variables. Indeed to perform simulations, the decomposition coefficients are sampled from the estimated Gaussian mixture, and the corresponding functions are constructed by multiplying the new coefficients with the basis functions.

In the next two sections, the methodology to characterize the uncertainty of dependent functional random variable linked to a covariate 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 analytical example in section 5.1, then the methodology is applied to a nuclear reliability example, in section 5.2.

2 Functional decomposition

Let us define the probability space $(\Omega, \mathcal{F}, P)$ and the functional random variables $f_1, \ldots, f_m : \Omega \times I \to \mathbb{R}$, where $I \subset \mathbb{R}$. $f_i(\omega, .) : I \to \mathbb{R}$, for $i \in \{1, \ldots, m\}$ 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 $f_{1,j}, \ldots, f_{m,j}$, $j \in \{1, \ldots, n\}$ is known. The corresponding outputs of $\mathcal{M}$, $y_j = \mathcal{M}(f_{1,j}, \ldots, f_{m,j})$, are also known. The functions are discretized on the points $t_1, \ldots, t_p$ of the interval I. The discretized version of the function $f_{i,j}$ is noted $\vec{f}_{i,j} \in \mathbb{R}^p$, $i \in \{1, \ldots, m\}$ and $j \in \{1, \ldots, n\}$, such that $f_{i,j}(t_k) = \vec{f}_{i,j,k}$, for $k \in \{1, \ldots, p\}$.

The objective of this section is to approximate simultaneously the m functional random variables $f_1, \ldots, f_m$ on a basis. The decomposition of a single functional random variable f_i, for $i \in \{1, \ldots, m\}$, is first presented. The sample functions $f_{i,1}, \ldots, f_{i,n}$ are approximated on a truncated basis $\left(\varphi_1^{(i)}, \ldots, \varphi_d^{(i)}\right)$ 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 $t \in \mathbb{R}$, $i \in \{1, \ldots, m\}$, $j \in \{1, \ldots, n\}$, $e^{(i)} = \frac{1}{n} \sum_{j=1}^{n} f_{i,j}$ is the mean function and $\alpha_{j,k}^{(i)}$ is the coefficient of the jth curve on the kth 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

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

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 express the functions on a spline basis. Then PCA can be applied to the coefficients of the functions on the spline basis. They also propose to apply PCA directly to the discretized functions. This second method is applied here. $F^{(i)}$ is the matrix of the n discretized functions such that $F_{k,j}^{(i)}=\vec{f}_{i,j,k}$. 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 on a single FPCA basis to handle the dependence between the functional random variables. In the following, this method is called SPCA. To this mean, the PCA decomposition is applied to the vectors $\vec{f}_j$ of concatenated discretized functions, such that

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

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:

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

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 hth 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 $X_{j,k} = \vec{f}_{i,j,k}$, $\forall j = 1, \ldots, n$ and $\forall k = 1, \ldots, p$. d steps are computed. Then, for $i = 1, \ldots, m$ and $\forall j = 1, \ldots, n$, the discretized sample functions can be approximated in this way:

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

The obtained vectors φ_h is then the hth basis function in the PLS decomposition and the hth predictor α_h is the vector of coefficients associated to the hth 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 PLS algorithm.

The choice of the decomposition depends on the studied case. If no covariate is known, SPCA is preferable in order to optimize the approximation of the functional variables. On the contrary, if a covariate is available, SPLS could be a better choice to add information about this covariate.

3 Probability density estimation

3.1 Gaussian Mixture model and EM algorithm

Let $\alpha_j = (\alpha_{j,1}, \ldots, \alpha_{j,d})$, $\forall j \in \{1, \ldots, n\}$ be the vectors of coefficients of the decomposition. The density of the sample of vectors $\alpha_1, \ldots, \alpha_n$ is estimated thanks to 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 f of the GMM is written $\forall \alpha \in \mathbb{R}^d$,

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

The parameters of the probability density function are estimated by the Expectation-Maximization algorithm (EM), introduced by Dempster et al. (1977). 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 unobserved data. In the case of GMM, the unobserved data 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:

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

In the EM algorithm, two steps are repeated until convergence. The Estimation 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. 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 practice. 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 $f_g : \alpha \mapsto \frac{e^{-(\alpha-\mu_g)^T \Sigma_g^{-1}(\alpha-\mu_g)/2}}{\sqrt{\det(2\pi\Sigma_g)}}$, 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 $\Sigma_k^{(0)}$, $\forall k \in \{1, \ldots, G\}$.
2. Expectation Step: Compute $z_{ik}^{(j)}$, $\forall k \in \{1, \ldots, G\}$, $\forall i \in \{1, \ldots, n\}$, thanks to equation (4).
3. Maximization Step: Compute $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$ and $\Sigma_k^{(j+1)}$, $\forall k \in \{1, \ldots, G\}$ thanks to 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 an 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.

3.2 Sparse Gaussian Mixture estimation

The total number N of parameters in the GM model 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

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

on the inverse of the covariance matrices. This algorithm is based on the method of Friedman et al. (2008) to estimate sparse inverse of 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 will follow a scheme close to the one of Krishnamurthy. However, we propose to apply directly the Lasso penalization on the covariance matrix thanks to 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 $g = 1, \ldots, G$:

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

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 $\forall i, j \in \{1, \ldots, n\}$,

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

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:

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

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 $\Sigma_k^{(0)}$, $\forall k \in \{1, \ldots, G\}$.
2. Expectation Step: Compute $z_{ik}^{(j)}$, $\forall k \in \{1, \ldots, G\}$, $\forall i \in \{1, \ldots, n\}$, thanks to equation (4).
3. Maximization Step: Compute $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$ and $\Sigma_k^{(j+1)}$, $\forall k \in \{1, \ldots, G\}$ thanks to equations (5), (6) and (7) respectively.
4. $\Sigma_k^{(j+1)} \leftarrow \text{argmin}_S \log\det S - \text{tr}(S^{-1}\Sigma_k^{(j+1)}) - \lambda \|P * S\|_1$.
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. The value of λ maximizing the computed log-likelihood is selected.

4 Criteria to assess the methodology quality

4.1 Criteria for the functional decomposition step

The functional decomposition of the functional variables is done conditionally to two objectives. The variables $f_1, \ldots, f_m$ must be approximated by the functional basis and the coefficients of the decomposition must be linked to the model output Y. 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 $\vec{f}_{1,j}, \ldots, \vec{f}_{m,j}$ for $j = 1, \ldots, n$ and their approximation by $\hat{\vec{f}}_{1,j}, \ldots, \hat{\vec{f}}_{m,j}$. The explained variance, denoted as criterion C_1 is then defined by this expression:

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

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

To quantify the link between the covariate and the coefficients, a linear model is estimated between these variables. The quality of the linear 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 $\bar{Y} = \sum_{j=1}^{n_t} Y_j$ is the output mean, and, for $j = 1, \ldots, n_t$, $\hat{Y}_j$ is the estimation of Y_j by the linear regression. In practice, the Q^2 can be computed by cross-validation. The second criterion is defined as the Q^2 computed by cross-validation and is denoted as C_2.

4.2 Criteria for the whole uncertainty quantification methodology

Three criteria have been chosen to assess the global methodology. First, the estimated probability distribution function of the coefficients is evaluated. To this mean, new samples of coefficients are simulated thanks to the estimated GMM. Their joint probability density function is compared to the one of a test sample of coefficients thanks to a multivariate goodness-of-fit test. The employed 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:

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

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

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

Finally, the ability of the methodology to reproduce the behaviour of the covariate is also tested. Similarly to the first criterion, 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

5.1 Analytical example

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.

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

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

where the random variables A_1, A_2 and A_3 follow uniform laws on respectively $[0, 03.05]$, $[0.05, 0.2]$ and $[2, 3]$, and with

$$h(t, A_3) = 0.15 \left(1 - \left|\frac{t - 100A_3}{60}\right|\right)$$

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

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 first decomposed on the SPCA and SPLS bases. The normalization factors are first compared. SPCA with 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. The

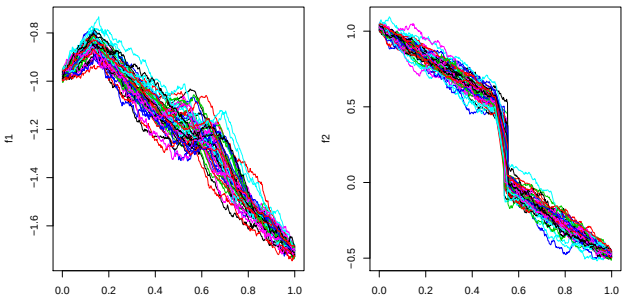

Figure 1: Samples of 600 realizations of f_1 (left) and f_2 (right).

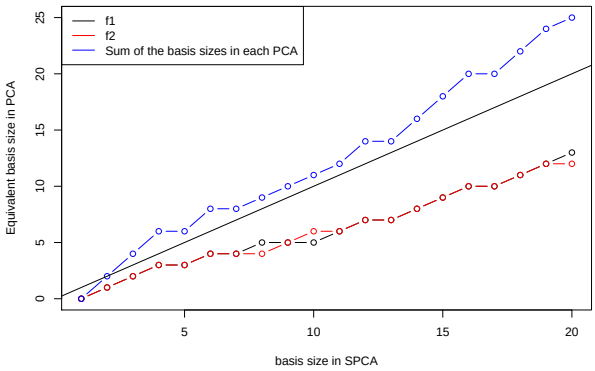

Figure 2: The maximal number of components of PCA of each variable such that the sum of the explained variances of these decompositions is lower than the explained variance of SPCA.

explained variances C_1 of SPCA and SPLS are compared with these of PCA and PLS respectively. The idea is to compare the explained variances of simultaneous and non-simultaneous decompositions for the same number of total components. 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 simultaneous decomposition. 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. 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 $y = x$ curve. If the sum of the number of components of each PLS (resp. PCA), the blue line, is over the $y = x$ line, 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 than PCA (resp. PLS) on f_1 and f_2 separately for an equal total number of components.

SPLS and SPCA are compared in Figures 4 and 5 on the criteria C_1 and C_2. In Figure 4, the percentage of explained variance as defined in equation (12) is drawn in function of the basis size in red for SPCA and in black for SPLS. The explained variance of SPCA is higher than the one of SPLS by definition. However, for basis with more than 8 components, the difference between the two explained variances becomes quite low. A linear 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 linear 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.

In the following, we focus on the SPLS decomposition. For different basis sizes, the probability density functions of the coefficients is estimated with the different estimation methods. 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

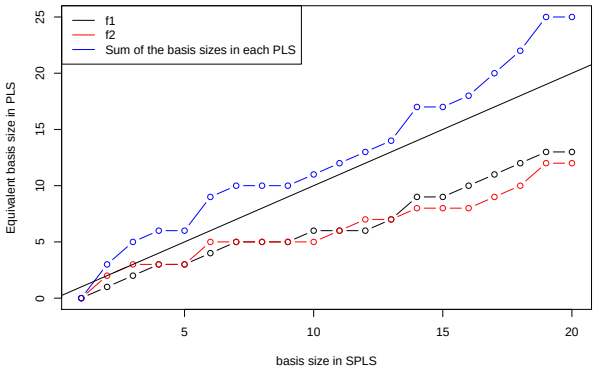

Figure 3: The maximal number of components of the PLS decomposition of each variable such that the sum of the explained variances of these decompositions is lower than the explained variance of SPLS.

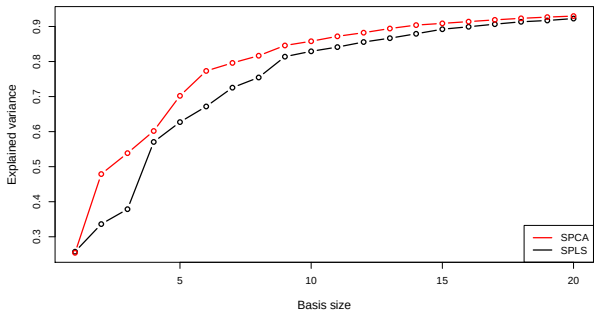

Figure 4: Explained variance by SPCA (in red) and SPLS (in black) (criterion C_1) as a function of the decomposition basis size.

toy_q2.pdf

Figure 5: Q^2 coefficient of the linear regression between the coefficients of SPCA (in red) or SPLS (in black) and the covariate (criterion C_2) as a function of the decomposition basis size.

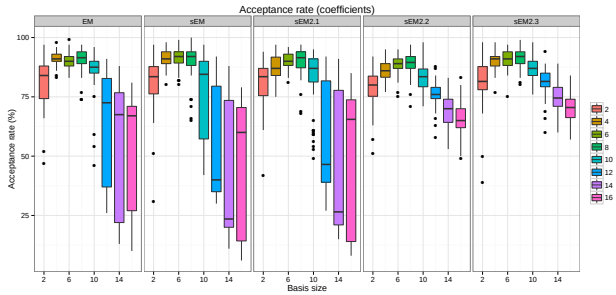

Figure 6: Boxplot of the acceptance rates for the goodness-of-fit test on the estimated coefficients density (criterion C_3^a) in function of the basis size and for each estimation algorithm.

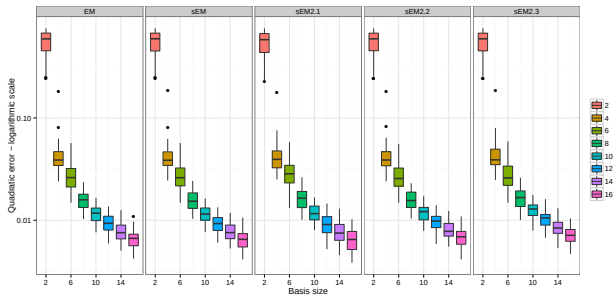

Figure 7: Boxplot of the mean square errors for the pointwise correlations (criterion C_3^b) in function of the basis size and for each estimation algorithm.

are computed with 10 test basis and 10 simulated 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 in 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 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

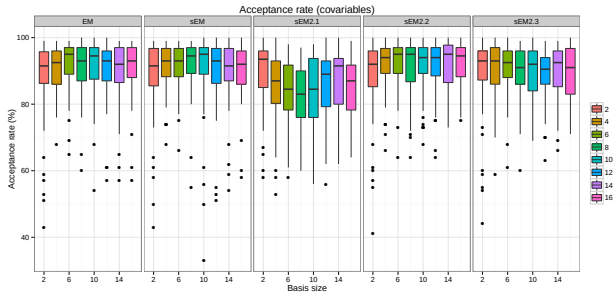

Figure 8: Boxplot of the acceptance rates for the goodness-of-fit test on the estimated covariates density (criterion C_3^c) in function of the basis size and for each estimation algorithm.

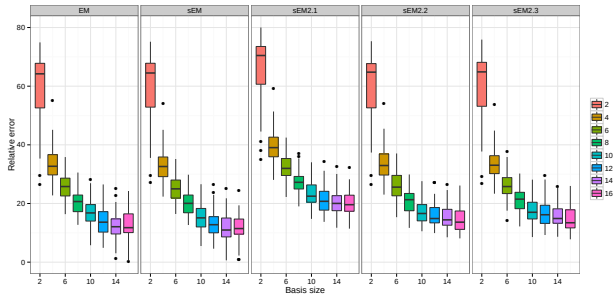

Figure 9: Boxplot of the relative approximation error between the estimated probability $\hat{p}$ and p in function of the basis size and for each estimation algorithm.

contrary, the acceptance rates of algorithm sEM2.1 vary much more, and even decrease in 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^b is over 90% with all algorithms except sEM2.1 for which it is about 85%. The criterion C_3^c is about 0.016 for bases with 8 functions and 0.012 for bases with 10 functions. As the values of both C_3^b and C_3^c criteria seem acceptable for bases with 8 or 10 functions, the chosen value for the basis size is here 8 or 10, depending on the threshold chosen for the C_3^a criterion. If 10 is chosen, the criterion C_3^a is a little worsened but the second one is improved.

In this analytical example, the global 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 quantification 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. Let us define the probability to estimate:

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

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 in function of the basis size and for each estimation algorithm. The estimation gives good results. For example, the medians of the errors are between 16 and 20% for bases of size 10. EM and sEM algorithms give slightly lower errors than other algorithms. The decrease of the error slows down for higher basis sizes.

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

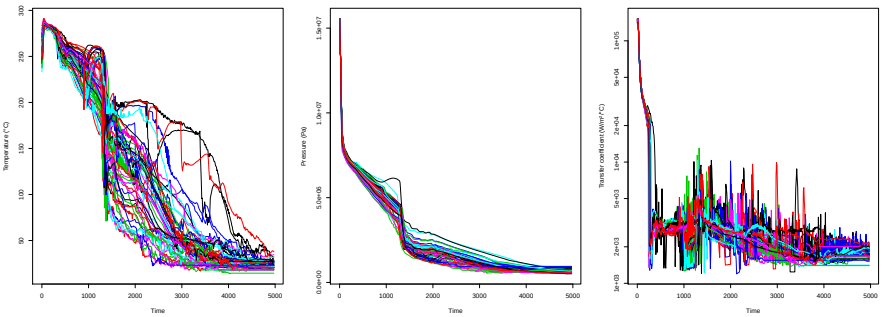

Figure 10: Samples of 400 curves of temperature (left panel), pressure (center) and transfer coefficient (right).

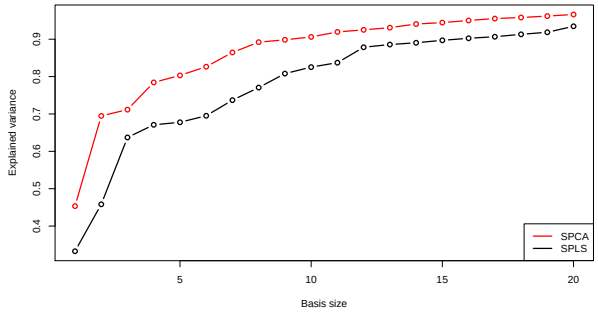

Figure 11: Explained variance by SPCA (in red) and SPLS (in black) (criterion C_1) as a function of the decomposition basis size.

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 hypothetic 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. Pessimistic values 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. 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 represents the variance explained by the SPLS (in red) and SPCA (in black) 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 linear regression 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 explains better than SPCA the link between the functional variables and the covariate. SPLS decomposition is used in the rest of the section.

The probability density function of the SPLS coefficients is estimated thanks to the EM, sEM and

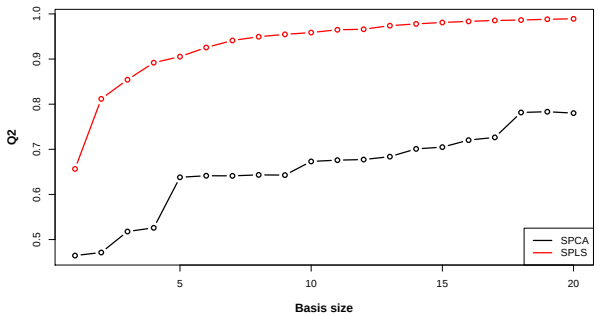

Figure 12: Q^2 of the linear regression between the coefficients of SPCA (in red) or SPLS (in black) and the covariate (criterion C_2) as a function of the decomposition basis size.

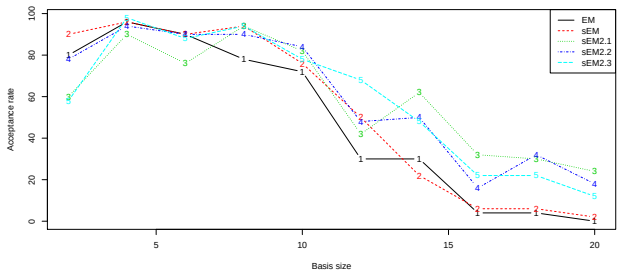

Figure 13: Acceptance rates for the goodness-of-fit test on the estimated coefficients density (criterion C_3^a) in function of the basis size.

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 in 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 in function of the basis size. Moreover, they are quite low for basis sizes over 6 or 8. From these two criteria, one could choose a decomposition basis with 10 functions, 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.

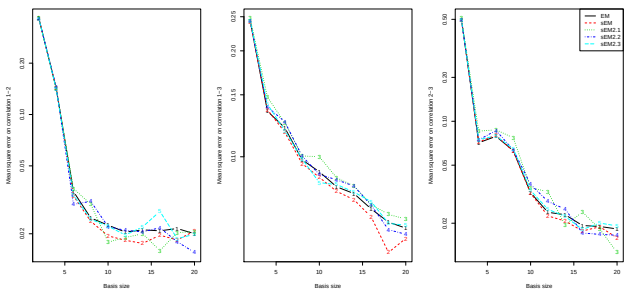

Figure 14: Mean square errors for the pointwise correlations (criterion C_3^b) in function of the basis size.

6 Conclusion

In this article, we have proposed a global methodology to quantify the uncertainties in a context of functional random variables. The distinctive features of the studied case are the facts that the functional variables are dependent and linked to a scalar or vectorial variable called covariate. An objective of the method is thus to preserve these features of the data during the modelling. 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 by the simultaneous principal component analysis or the developed simultaneous partial least squares decomposition. The latter one has the advantage to maximize the link 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 Gaussian mixture model with sparse covariance matrices.

The uncertainty quantification methodology has been successfully applied to an analytical example with two functional random variables and to a nuclear assessment test case. In both presented test examples, the SPLS algorithm has been shown to better explain the link between the functional variables and 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 thanks to the probability density function estimated in the methodology. Finally, the presented method can be used for uncertainty quantification in computed codes. 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., Mara, T., Denis-Vidal, L., and Goffart, J. (2013). Uasa of complex models: Coping with dynamic and static inputs. In *7th International Conference on Sensitivity Analysis of Model Output, SAMO 2013*.

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

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

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

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

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

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

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

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

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

Pearson, K. (1901). 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.

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

Saltelli, A., Chan, K., and Scott, E. (2000). *Sensitivity analysis*. Wiley series in probability and statistics. Wiley.

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

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

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

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

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