Uncertainty quantification and visualization for functional random variables

MascotNum Workshop 2014

S. Nanty [1,3] C. Helbert [2] A. Marrel [1] N. Pérot[1] C. Prieur [3]

[1]CEA, DEN/DER/SESI/LSMR, F-13108, Saint-Paul-lez-Durance, France

[2]Ecole Centrale de Lyon, France

[3]Universite Joseph Fourier and INRIA, France

April 23rd 2014

Introduction

- Identify/characterize the statistical properties of functional random variables.
- The variables are dependent and linked to a scalar (or vectorial) covariate.
- Propose a methodology of uncertainty characterization in order to:
  - get an estimate of the joint probability density function of the variables,
  - simulate new samples according to the estimated distribution,
  - adapt visualization tools to identify uncertainty characteristics of dependent functional variables.

Problem description

- Let $f_1, \dots, f_m : I \times \Omega \to \mathbb{R}$ be dependent functional random variables.
- Let Y be a random variable, called covariate.
- Let $\mathcal{M}$ be a computer code/simulator such that

$$Y = \mathcal{M}(f_1, \dots, f_m).$$

- Let f_j^i be the i^{th} realization of the j^{th} functional random variable, for $1 \leq i \leq n,\ 1 \leq j \leq m$.

Proposed methodology

- Two main steps:
  1. Decomposition on a reduced functional basis, taking into account the covariate
  2. Modeling of the probability density function of the decomposition coefficients

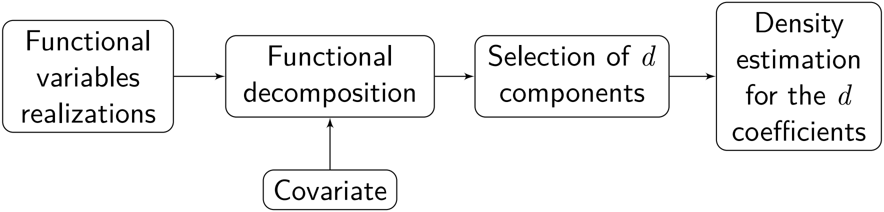


Table of Contents

Decomposition on a functional basis

Definition

Let $f : I \to \mathbb{R}$, $x \in I$.

$$f(x) = \sum_{k=1}^{+\infty} \alpha_k \phi_k(x)$$

- α_k coefficients,
- ϕ_k basis functions

Decomposition on a functional basis

Definition

Let $f : I \to \mathbb{R}$, $x \in I$.

$$\hat{f}(x) = \sum_{k=1}^{d} \alpha_k \phi_k(x)$$

- α_k coefficients,
- ϕ_k basis functions,
- d basis size

Partial Least Squares regression

- Let X $(n \times p)$ and Y $(n \times q)$ data matrices of respectively observable and predicted variables.
- X and Y are centered and standardized.
- Principle: linear regression between the projections of X and Y in a new space, called latent variables, whose correlation is maximal.

Algorithm of PLS regression [Wold, 1975]

- Initialization: $X_0 = X$, $Y_0 = Y$
- At each step h, we are seeking for the latent variables $\alpha_h = X_{h-1}u_h$ and $\omega_h = Y_{h-1}v_h$ solutions of

$$\max_{\|u_h\|=\|v_h\|=1} \operatorname{cov}(X_{h-1}u_h, Y_{h-1}v_h).$$

- "Deflation": $X_h = X_{h-1} - \alpha_h \phi_h^T$, with $\phi_h = X_{h-1}\alpha_h / (\alpha_h^T \alpha_h)$

Partial Least Squares regression

- Let X $(n \times p)$ and Y $(n \times q)$ data matrices of respectively observable and predicted variables.
- X and Y are centered and standardized.
- Principle: linear regression between the projections of X and Y in a new space, called latent variables, whose correlation is maximal.

Algorithm of PLS regression [Wold, 1975]

- Initialization: $X_0 = X,\ Y_0 = Y$
- At each step h, we are seeking for the latent variables $\alpha_h = X_{h-1}u_h$ and $\omega_h = Y_{h-1}v_h$ solutions of

$$\max_{\|u_h\|=\|v_h\|=1} \operatorname{cov}(X_{h-1}u_h, Y_{h-1}v_h).$$

- "Deflation": $X_h = X_{h-1} - \alpha_h \phi_h^T$, with $\phi_h = X_{h-1}\alpha_h / (\alpha_h^T \alpha_h)$

Partial Least Squares decomposition

- It can be deduced from the deflation step that X can be written as follows:

$$X = A\Phi^T + \epsilon$$

where the column vectors of A and Φ are respectively α_h and ϕ_h and ϵ are the residuals.

- Let the column vectors of X be functions discretized on p points and Y be the covariate.
  - $\Rightarrow$ A is the matrix of coefficients of the decomposition.
  - $\Rightarrow$ Φ is the matrix of basis functions.

- Basis functions are fitted to data, and
- adjusted to maximize the correlation between the functions and the covariate.

Partial Least Squares decomposition

- It can be deduced from the deflation step that X can be written as follows:

$$X = A\Phi^T + \epsilon$$

  where the column vectors of A and Φ are respectively α_h and ϕ_h and ϵ are the residuals.

- Let the column vectors of X be functions discretized on p points and Y be the covariate.
  - ⇒ A is the matrix of coefficients of the decomposition.
  - ⇒ Φ is the matrix of basis functions.

- Basis functions are fitted to data, and
- adjusted to maximize the correlation between the functions and the covariate.

Partial Least Squares decomposition

- It can be deduced from the deflation step that X can be written as follows:

$$X = A\Phi^T + \epsilon$$

where the column vectors of A and Φ are respectively α_h and ϕ_h and ϵ are the residuals.

- Let the column vectors of X be functions discretized on p points and Y be the covariate.
  - $\Rightarrow$ A is the matrix of coefficients of the decomposition.
  - $\Rightarrow$ Φ is the matrix of basis functions.

- Basis functions are fitted to data, and
- adjusted to maximize the correlation between the functions and the covariate.

Simultaneous PLS decomposition

- Objective: extend PLS decomposition to deal with multiple functional dependent variables simultaneously
- We suppose that functions $f_1 \dots f_m$ are correlated and have common reduction directions.
- Let $t_1 < \cdots < t_p \in I$
- Let $\mathbf{f}_i = (f_i(t_1), \dots, f_i(t_p))$ be the discretized version of f_i, $i = 1, \dots, m$.
- Let each column vector of X be:

$$(\mathbf{f}_1, \dots, \mathbf{f}_m) \in \mathbb{R}^{dm}$$

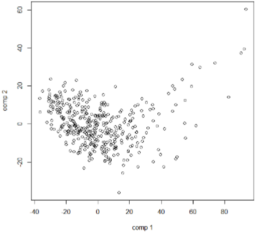
- Simultaneous PLS decomposition consists in applying the PLS decomposition to the previously defined matrix X.

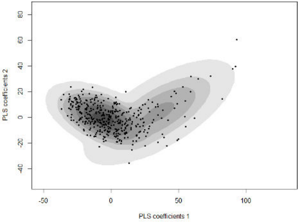
  $\rightarrow$ **SPLS decomposition**

Objectives

- Estimate the probability density function (pdf) of d coefficients from SPLS decomposition
- High dimension: $d > 10$
  - $\Rightarrow$ kernel density estimation not adapted

$\rightarrow$ Solution: **Gaussian mixture model**





Gaussian Mixture

- Probability density function of a Gaussian mixture:

$$g(\alpha|\mu_1, \Sigma_1, \ldots, \mu_G, \Sigma_G) = \sum_{k=1}^{G} \tau_k \phi(\alpha|\mu_k, \Sigma_k), \ \forall \alpha \in \mathbb{R}^d$$

  - G clusters
  - n sample points
  - ϕ: Gaussian probability density function
  - $\tau_k, \ \mu_k, \ \Sigma_k$: proportion, mean and covariance matrix of cluster k

- Advantages / drawbacks
  - + Fast algorithm for parameter estimation
  - + Very fast simulation of a new realization
  - + Can be used in dimension $d > 10$
  - − parametric model: modeling hypothesis
  - − Number of clusters to be determined

Gaussian Mixture

- Probability density function of a Gaussian mixture:

$$g(\alpha|\mu_1, \Sigma_1, \ldots, \mu_G, \Sigma_G) = \sum_{k=1}^{G} \tau_k \phi(\alpha|\mu_k, \Sigma_k), \ \forall \alpha \in \mathbb{R}^d$$

  - G clusters
  - n sample points
  - ϕ: Gaussian probability density function
  - $\tau_k, \ \mu_k, \ \Sigma_k$: proportion, mean and covariance matrix of cluster k

- Advantages / drawbacks
  - \+ Fast algorithm for parameter estimation
  - \+ Very fast simulation of a new realization
  - \+ Can be used in dimension $d > 10$
  - − parametric model: modeling hypothesis
  - − Number of clusters to be determined

EM algorithm

- Expectation-Maximization algorithm (EM) [Dempster et al., 1977] estimates the parameters of the Gaussian mixture model.
- Let us introduce z_{ik}, the probability of the i^{th} point to be in the k^{th} cluster.

Expectation Minimization algorithm:

1. Initialize parameters $\tau_k^{(0)}$, $\mu_k^{(0)}$ et $\Sigma_k^{(0)}$
2. Expectation Step: Compute $z_{ik}^{(j)}$
3. Maximization Step: Compute $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$, $\Sigma_k^{(j+1)}$
4. Repeat steps $2-3$ until convergence

Number of parameters reduction

- Total number of Gaussian mixture parameters: $N_T = G - 1 + Gd + G\frac{d(d+1)}{2}$
- G: number of clusters in the model
- N_T increases quickly with the dimension d

$\rightarrow$ Solution: **sparse covariance matrices estimation**

Two methods

- sEM method: penalizing the inverses of covariance matrices [Krishnamurthy, 2011]
- sEM2 method: penalizing the covariance matrices

sEM method

Penalizing the inverses of covariance matrices

- A **lasso penalization** on the inverses of the covariance matrices is added in the maximization step:

$$\hat{\Sigma}_k = \text{argmax}_{\Sigma_k}\,(\ell) \quad \dashrightarrow \quad \hat{\Sigma}_k = \text{argmax}_{\Sigma_k}\left(\ell - \lambda \|\Sigma_k^{-1}\|_1\right)$$

- $\|M\|_1 = \sum_{i,j=1}^{p} M_{i,j}$.
- The penalization parameter λ is chosen bycross-validation.

- The penalized maximization is solved by [Friedman et al., 2008] coordinate descent-based algorithm.

sEM method

Penalizing the inverses of covariance matrices

sEM algorithm [Krishnamurthy, 2011]

1. Initialize parameters $\tau_k^{(0)}$, $\mu_k^{(0)}$ et $\Sigma_k^{(0)}$
2. Expectation Step: Compute $z_{ik}^{(j)}$
3. Maximization Step: Compute $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$
4. $\Sigma_k^{(j+1)} \leftarrow \text{argmax}_\Sigma \left(\ell - \lambda \left\|\Sigma^{-1}\right\|_1\right)$
5. Repeat steps $2-4$ until convergence

sEM2 method

Penalizing the covariance matrices

- A **lasso** penalization on the covariance matrices is added in the maximization step:

$$\hat{\Sigma}_k = \text{argmax}_{\Sigma_k} \left(\ell - \lambda \| P * \Sigma_k \|_1 \right)$$

- $*$ stands for Hadamard product.
- P: penalization matrix.
- The penalization parameter λ is chosen by cross-validation.
- The penalized maximization is solved by [Wang, 2013] coordinate descent-based algorithm.

sEM2 method

Penalizing the covariance matrices

- Several proposed matrices P:
  - **sEM2.1**: Equal weights to all matrix elements. All elements are penalized in the same way.

$$P_{ij} = 1$$

  - **sEM2.2**: Diagonal elements are not penalized. All others are penalized equally.

$$P_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

  - **sEM2.3**: The lower the off-diagonal element, the more penalized. Diagonal elements are not penalized.

$$P_{ij} = \begin{cases} \frac{1}{\Sigma_{ij}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

sEM2 method

Penalizing the covariance matrices

- Several proposed matrices P:
  - **sEM2.1**: Equal weights to all matrix elements. All elements are penalized in the same way.
$$P_{ij} = 1$$
  - **sEM2.2**: Diagonal elements are not penalized. All others are penalized equally.
$$P_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$
  - **sEM2.3**: The lower the off-diagonal element, the more penalized. Diagonal elements are not penalized.
$$P_{ij} = \begin{cases} \frac{1}{\Sigma_{ij}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

sEM2 method

Penalizing the covariance matrices

- Several proposed matrices P:
  - **sEM2.1**: Equal weights to all matrix elements. All elements are penalized in the same way.

    $$P_{ij} = 1$$

  - **sEM2.2**: Diagonal elements are not penalized. All others are penalized equally.

    $$P_{ij} = \begin{cases} 1 & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

  - **sEM2.3**: The lower the off-diagonal element, the more penalized. Diagonal elements are not penalized.

    $$P_{ij} = \begin{cases} \frac{1}{\Sigma_{ij}} & \text{if } i \neq j \\ 0 & \text{if } i = j \end{cases}$$

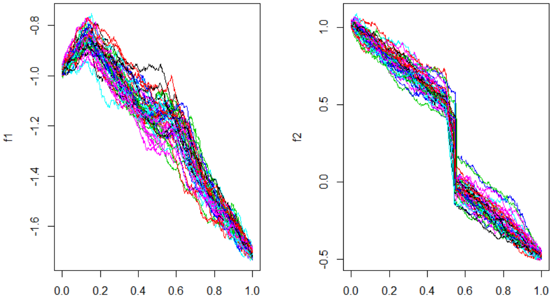
sEM2 method

Penalizing the covariance matrices

sEM2 algorithm

1. Initialize parameters $\tau_k^{(0)}$, $\mu_k^{(0)}$ et $\Sigma_k^{(0)}$
2. Expectation Step: Compute $z_{ik}^{(j)}$
3. Maximization Step: Compute $\tau_k^{(j+1)}$, $\mu_k^{(j+1)}$
4. $\Sigma_k^{(j+1)} \leftarrow \text{argmax}_\Sigma \left(\ell - \lambda \left\| P * \Sigma \right\|_1\right)$
5. Repeat steps $2 - 4$ until convergence

Illustration on an analytical example



- 2 temporal functional random variables f_1 et f_2 depending on three random variables a_1, a_2, a_3.
- a_1, a_2, a_3 have uniform distributions.
- Let define the covariate

$$Y(a_1, a_2, a_3) = \int_0^1 (f_1(t, a_1, a_2, a_3) + f_2(t, a_1, a_2, a_3))\, dt.$$

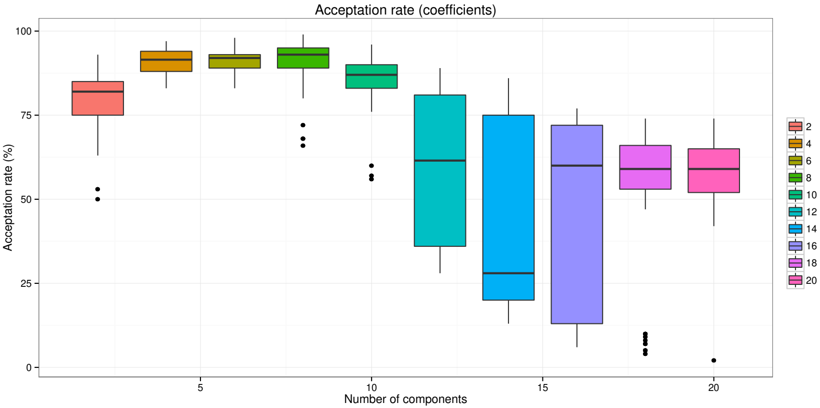
Illustration on an analytical example

- Hypothesis
  - Learning dataset: 600 curves
  - Test dataset: 1000 curves
  - SPLS decomposition + Gaussian mixture model
  - Optimal number G^* of clusters chosen with Bayesian Information Criterion (BIC)
- Proposed criteria to select the basis size d and assess the quality of the characterization method:
  - **Criterion C1**: Goodness-of-fit of estimated coefficients pdf and the real pdf with1 [Fromont et al., 2012] test.
  - **Criterion C2**: Goodness-of-fit of estimated covariate pdf and the pdf computed with known covariates with Kolmogorov-Smirnov (KS) test.
  - **Criterion C3**: Relative mean square between correlation on estimated functions and realizations of the variables.

$\rightarrow$ First step: use of EM algorithm

Illustration on an analytical example

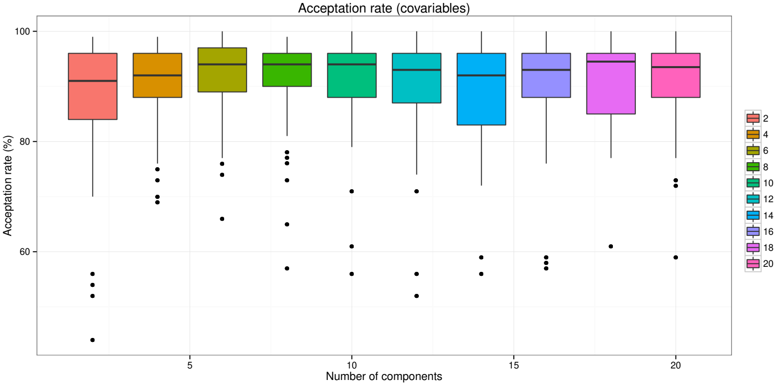
Criterion C1: Comparison of coefficients densities



- Maximal median at $d = 8$ components.
- After $d = 8$, model quality decreases.

Illustration on an analytical example

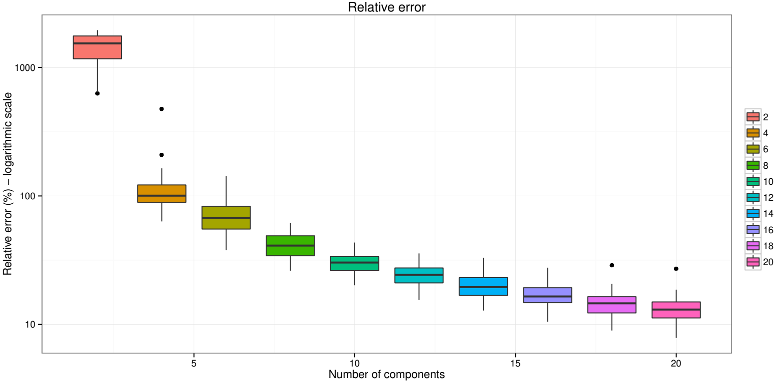
Criterion C2: Comparison of covariates densities



- Maximal median at $d = 6, 8, 10, 18$ components.
- Low variance for $d = 8$.
- Very close acceptance rates for all basis sizes.

Illustration on an analytical example

Criterion C3: Comparison of correlations between variables



- The correlation decrease is very even.
- Relative error is about 40% for $d = 8$.

Illustration on an analytical example: conclusions

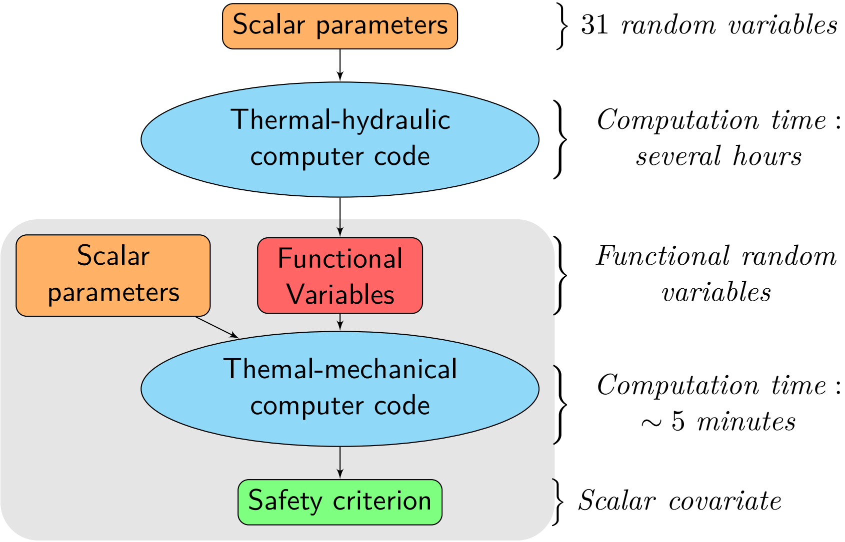
- Based on the three criteria → optimal basis size $d^* = 8$:
  - Good acceptance rates are obtained with EM algorithm.
  - The relative errors on correlation are still quite high.

- The same criteria have been computed for other estimation algorithms → similar results obtained (same d^* and criteria values).

- For the analytical example, the EM algorithm seems to be the best choice (efficient, easy and fast): as the number of parameters is quite low in this example ($n = 89$ for $d = 8$), the use of sparse algorithms does not improve the estimation.

- In practice, if no test basis is available, criteria C1, C2 and C3 are computed by cross-validation.

A nuclear safety test case (1)



A nuclear safety test case (2)

Dataset:

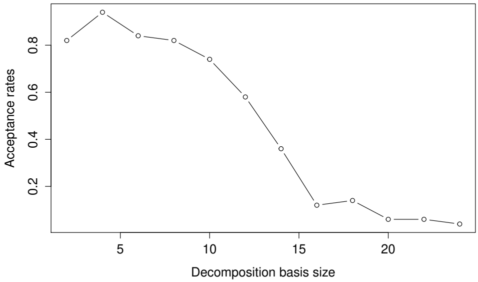
- 3 functional random variables depending on time
- Scalar covariate: a safety criterion
- Learning sample: 400 samples
- Logarithmic transformation of the sample (positivity constraint)

Methodology:

- SPLS decomposition + Gaussian mixture model + EM algorithm
- Optimal G^* determined by BIC
- Criteria C1, C2 and C3 computed by cross-validation
- Optimal d^* chosen by the analysis of the three criteria

A nuclear safety test case (3)

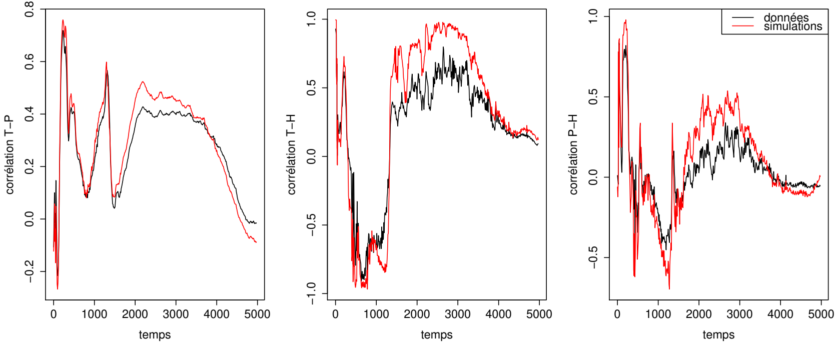
- Criterion 1:
    - optimal $d^* = 4$
    - acceptance rates under 80% for $d > 8$
    - fast decrease of acceptance rates for $d \geqslant 10$



- Criterion 2: low acceptance rates for all basis sizes

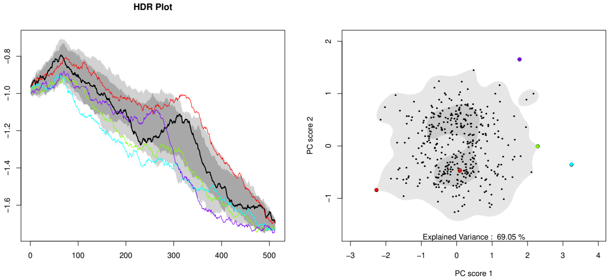
A nuclear safety test case (4)

- Criterion 3: quite good approximation of functional variable correlations for $d^* = 4$.



Visualization: High Density Region boxplot (HDR)

- Proposed by [Hyndman and Shang, 2010] and based on
  - Principal Component Analysis
  - First two basis functions selected
  - Kernel density estimation

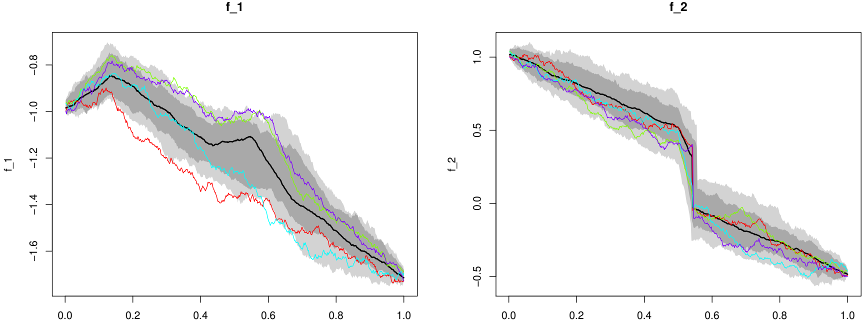


- Application on the analytical example:
  - Black curve: functional median
  - Colored curves: outliers
  - Dark (resp. light) gray zone: 50% (resp. 95%) highest density region

Visualization: Modified HDR boxplot

- Combination of the HDR boxplot and our proposed characterization methodology (SPLS + Gaussian mixture model)
  - ⇒ Simultaneous visualization of multiple functions
  - ⇒ Taking into account a covariate
  - ⇒ Decomposition on higher basis

- Illustration on the analytical example:



Conclusion and perspectives

- Development of a global methodology to simultaneously characterize dependent functional random variables linked to a covariate.

→ **Simultaneous PLS decomposition + Gaussian mixture with sparse covariance matrices**

  ⇒ Estimation of probabilities for the variables to exceed a threshold.
  ⇒ Simulation according to the estimated pdf.
  ⇒ Visualization of the uncertainty of the variables.

- Different proposed criteria to assess the methodology efficiency:
  - Application on an analytical example: good results
  - Application on a nuclear safety test case: functions and correlations quite well reproduced but the covariate pdf not well fitted

Perspectives:

- Computing probabilities and quantiles to exceed a threshold.
- Using this methodology to run uncertainty propagation and sensitivity analysis studies.

References

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), pages 1–38.

Friedman, J., Hastie, T., and Tibshirani, R. (2008).
Sparse inverse covariance estimation with the graphical lasso.
Biostatistics, 9(3):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 23.1–23.22.

Hyndman, R. J. and Shang, H. L. (2010).
Rainbow plots, bagplots, and boxplots for functional data.
Journal of Computational and Graphical Statistics, 19(1):29–45.

Krishnamurthy, A. (2011).
High-dimensional clustering with sparse gaussian mixture models.

Wang, H. (2013).
Coordinate descent algorithm for covariance graphical lasso.
Statistics and Computing, pages 1–9.

Wold, H. (1975).
Path models with latent variables: The nipals approach.

Appendix

Analytical example definition

$$\begin{aligned} f_1(t, a_1, a_2, a_3) &= 0.8 a_2 BB(t) + a_1 + c_1(t) + h(t) \\ f_2(t, a_1, a_2, a_3) &= a_2 BB(t) + a_1 + c_2(t) \end{aligned}$$

with

$$a_1 \sim \mathcal{U}(0, 0.05) \ ; \ a_2 \sim \mathcal{U}(0.05, 0.2) \ ; \ a_3 \sim \mathcal{U}(2, 3)$$

$$\begin{aligned} c_1(t) &= \begin{cases} t - 1 & \text{if } t < \frac{70}{512} \\ \frac{372}{512} - t & \text{otherwise} \end{cases} \\ c_2(t) &= \begin{cases} 1 - t & \text{if } t < 0.5 \\ \frac{64}{5a_3} - 0.5t & \text{if } 0.5 < t < 0.5 + \frac{10a_3}{512} \\ 0.5 - t & \text{otherwise} \end{cases} \\ h(t) &= 0.15 \left(1 - \left| \frac{t - 100 a_3}{60} \right| \right) \end{aligned}$$