

Nonlinear dimensionality reduction for functional computer code modelling

Benjamin Auder

CEA - UPMC

2010, September 22

Thesis since 02/2008

PhD supervisors : Gérard Biau (UPMC)

Bertrand Iooss (EDF)

Industrial context

Framework : life span of reactor vessels.

→ Several sequences of accidents can occur.

Goal = estimate their probabilities of occurrence.

Industrial context

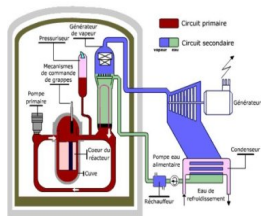
Framework : life span of reactor vessels.

→ Several sequences of accidents can occur.

Goal = estimate their probabilities of occurrence.

Methodology

Modeling



Industrial context

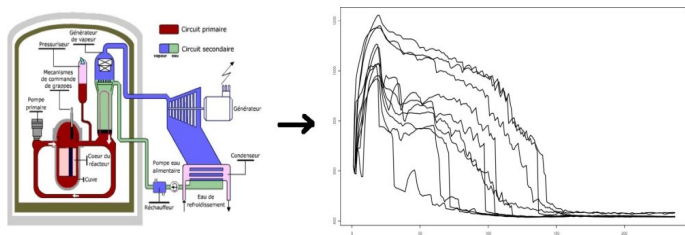
Framework : life span of reactor vessels.

→ Several sequences of accidents can occur.

Goal = estimate their probabilities of occurrence.

Methodology

Modeling → Simulation



Industrial context

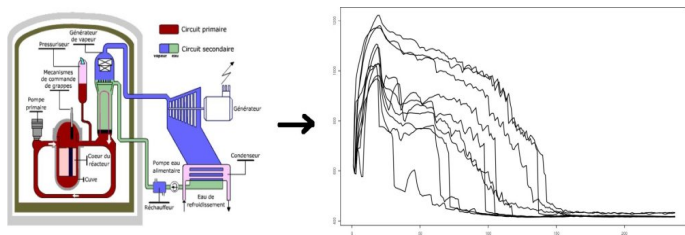
Framework : life span of reactor vessels.

→ Several sequences of accidents can occur.

Goal = estimate their probabilities of occurrence.

Methodology

Modeling → Simulation → Computation.



→ Sensitivity analysis,
uncertainty propagation
...

Industrial context

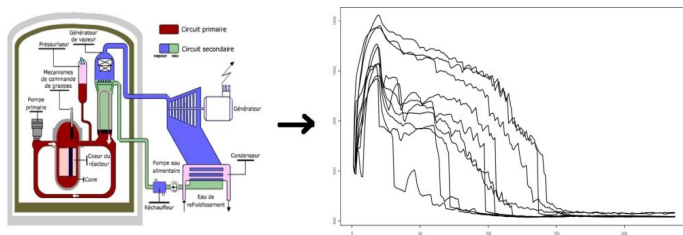
Framework : life span of reactor vessels.

→ Several sequences of accidents can occur.

Goal = estimate their probabilities of occurrence.

Methodology

Modeling → Simulation → Computation.



→ Sensitivity analysis,
uncertainty propagation
...

Improve the simulation stage to allow accurate computations

In my laboratory (CEA) ...

Thermal-hydraulic code CATHARE = time consuming

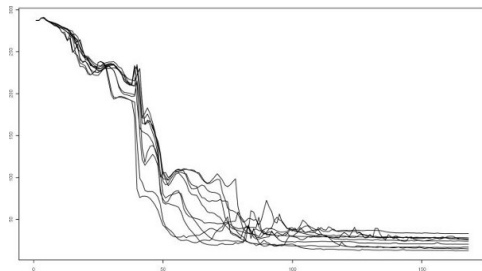


FIG.: Temperature transients.

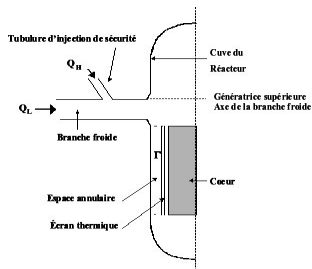


FIG.: Modelized zone.

In my laboratory (CEA) ...

Thermal-hydraulic code CATHARE = time consuming

- "black box" ;

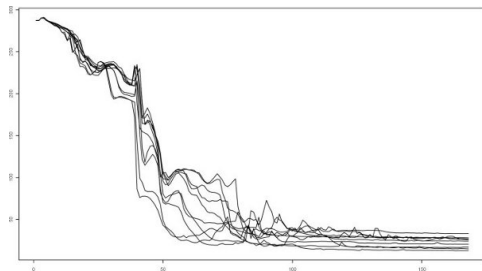


FIG.: Temperature transients.

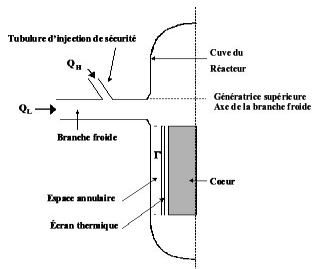


FIG.: Modelized zone.

In my laboratory (CEA) ...

Thermal-hydraulic code CATHARE = time consuming

- "black box" ;
- uncertain input parameters.

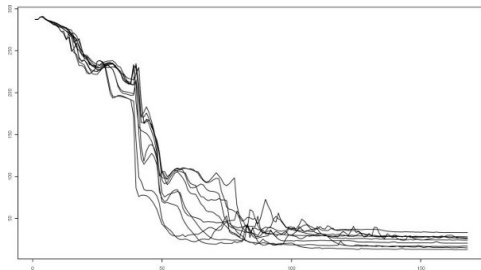


FIG.: Temperature transients.

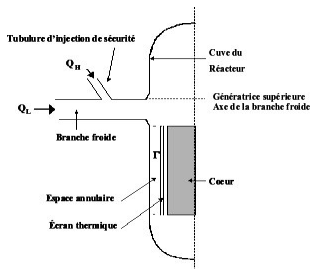


FIG.: Modelized zone.

In my laboratory (CEA) ...

Thermal-hydraulic code CATHARE = time consuming

- "black box" ;
- uncertain input parameters.

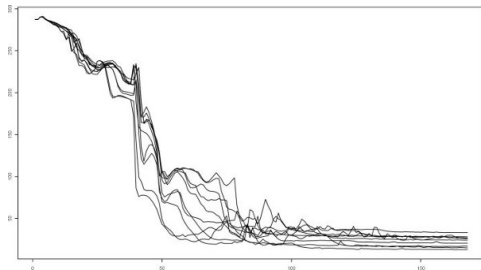


FIG.: Temperature transients.

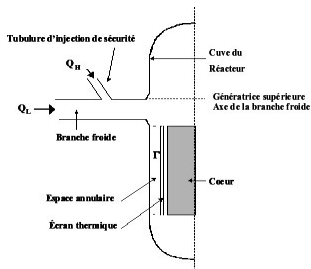


FIG.: Modelized zone.

"Speed up" the execution of CATHARE code

In my laboratory (CEA) ...

Thermal-hydraulic code CATHARE = time consuming

- "black box" ;
- uncertain input parameters.

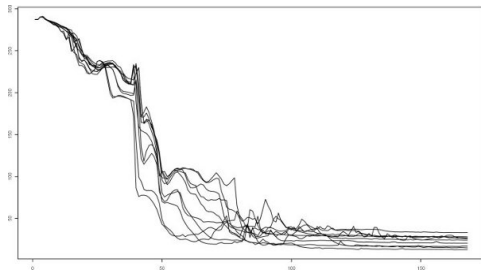


FIG.: Temperature transients.

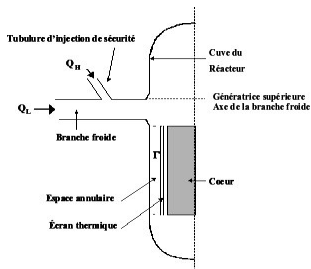


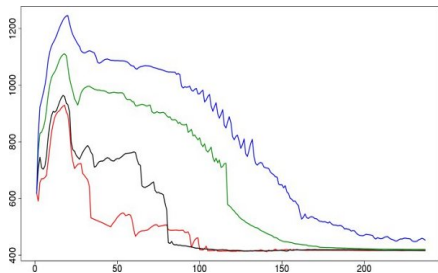
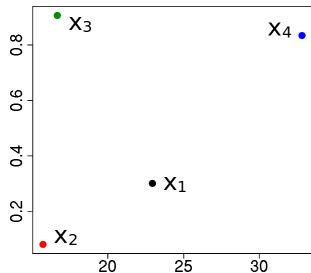
FIG.: Modelized zone.

"Speed up" the execution of CATHARE code = statistical modeling

Mathematical formulation

n known couples $(x_i, y_i) =$ inputs-outputs of a **very slow code** :

- Inputs $x_i \in \mathbb{R}^p =$ initial state of physical system ;
- Outputs $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$ evolutions of parameters.



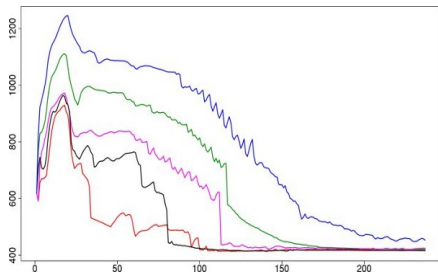
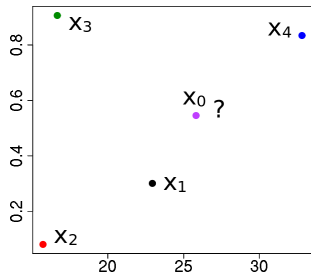
Mathematical formulation

n known couples $(x_i, y_i) =$ inputs-outputs of a **very slow code** :

- Inputs $x_i \in \mathbb{R}^p =$ initial state of physical system ;
- Outputs $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$ evolutions of parameters.

Goal = **prediction** of functional data :

$$y^{\text{new}} \simeq \varphi(x^{\text{new}}).$$



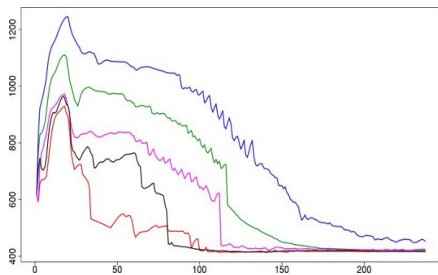
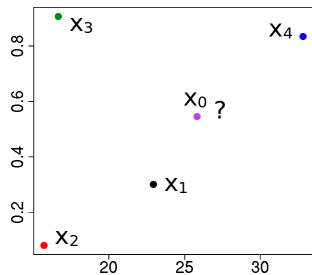
Mathematical formulation

n known couples $(x_i, y_i) =$ inputs-outputs of a **very slow code** :

- Inputs $x_i \in \mathbb{R}^p =$ initial state of physical system ;
- Outputs $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$ evolutions of parameters.

Goal = **prediction** of functional data :

$$y^{\text{new}} \simeq \varphi(x^{\text{new}}).$$



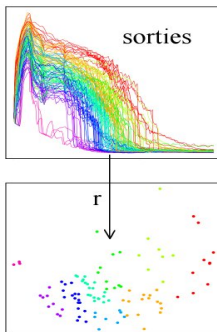
Statistical learning "regression" $\mathbb{R}^p \rightarrow \mathcal{C}([a, b], \mathbb{R})$

Back to the "simple" case of $y_i \in \mathbb{R}^d$

Back to the "simple" case of $y_i \in \mathbb{R}^d$

① dimensionality reduction :

$$r : \mathcal{C}([a, b], \mathbb{R}) \rightarrow \mathbb{R}^d \text{ (representation) ;}$$



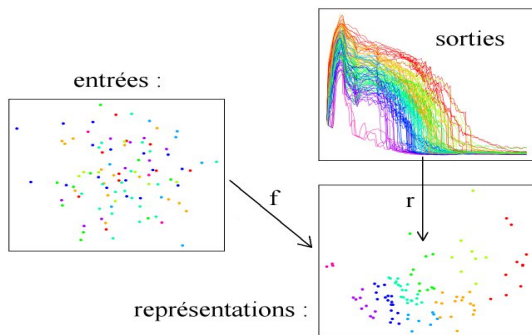
Back to the "simple" case of $y_i \in \mathbb{R}^d$

- 1 dimensionality reduction :

$$r : \mathcal{C}([a, b], \mathbb{R}) \rightarrow \mathbb{R}^d \text{ (representation);}$$

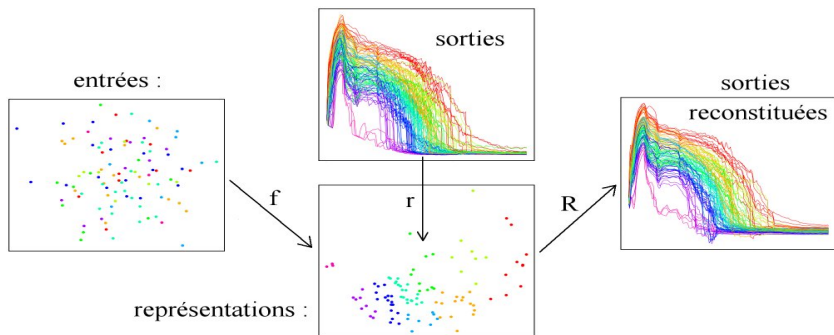
- 2 statistical learning :

$$f : \mathbb{R}^p \text{ (inputs)} \rightarrow \mathbb{R}^d \text{ (reduced outputs);}$$



Back to the "simple" case of $y_i \in \mathbb{R}^d$

- 1 dimensionality reduction :
 $r : \mathcal{C}([a, b], \mathbb{R}) \rightarrow \mathbb{R}^d$ (representation) ;
- 2 statistical learning :
 $f : \mathbb{R}^p$ (inputs) $\rightarrow \mathbb{R}^d$ (reduced outputs) ;
- 3 output space parametrization :
 $R : \mathbb{R}^d \rightarrow \mathcal{C}([a, b], \mathbb{R})$ (reconstruction).



Back to the "simple" case of $y_i \in \mathbb{R}^d$

- 1 dimensionality reduction :

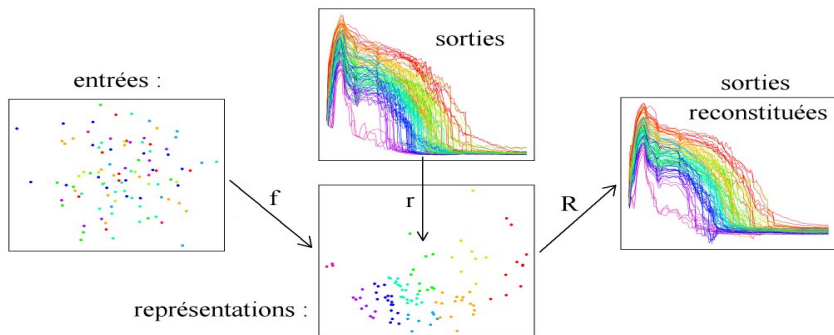
$$r : \mathcal{C}([a, b], \mathbb{R}) \rightarrow \mathbb{R}^d \text{ (representation);}$$

- 2 statistical learning :

$$f : \mathbb{R}^p \text{ (inputs)} \rightarrow \mathbb{R}^d \text{ (reduced outputs);}$$

- 3 output space parametrization :

$$R : \mathbb{R}^d \rightarrow \mathcal{C}([a, b], \mathbb{R}) \text{ (reconstruction).}$$



State of art

(More or less) classical methods

- Functional linear regression :
Faraway, 1997 ; Ramsay & Silverman, 2005, ...

State of art

(More or less) classical methods

- Functional linear regression :
Faraway, 1997 ; Ramsay & Silverman, 2005, ...
- Decomposition on an orthonormal basis, then learning of d -dimensional coefficients :
Chiou et al., 2004 ; Govaerts & Noël, 2005 ;
Bayarri et al., 2007 ; Marrel, 2008 ; Monestiez & Nerini, 2009

State of art

(More or less) classical methods

- Functional linear regression :
Faraway, 1997 ; Ramsay & Silverman, 2005, ...
- Decomposition on an orthonormal basis, then learning of d -dimensional coefficients :
Chiou et al., 2004 ; Govaerts & Noël, 2005 ;
Bayarri et al., 2007 ; Marrel, 2008 ; Monestiez & Nerini, 2009

(New) goal : minimize the representation dimension d , to

- simplify the model ;
- avoid overfitting,

while keeping good performances.

1 "Stepwise" dimension reduction

- Local PCA Manifold Learning (Zhan et al. 2008)
- Riemannian Manifold Learning (Lin et al. 2006)

2 Applications

1 "Stepwise" dimension reduction

- Local PCA Manifold Learning (Zhan et al. 2008)
- Riemannian Manifold Learning (Lin et al. 2006)

2 Applications

Overview

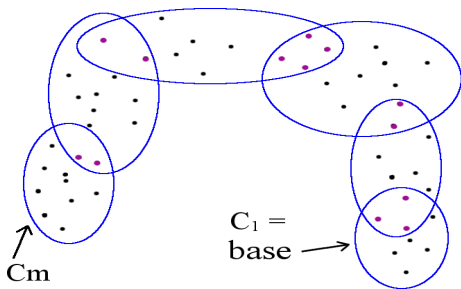
Goal : transform y_i into $z_i \in \mathbb{R}^d$ with "small" d .

Overview

Goal : transform y_i into $z_i \in \mathbb{R}^d$ with "small" d .

Local PCA Manifold Learning
(Zhan et al., 2008);

Clé : "Traversal Sequence of
Local Neighborhoods"



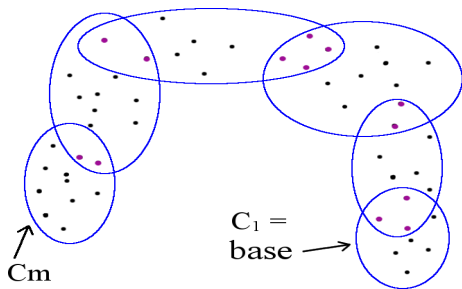
- Series $C_1, \dots, C_m \subset \{1, \dots, n\}$ covering $\{1, \dots, n\}$, with non-empty intersections $C_i \cap C_j, \forall i > j$;

Overview

Goal : transform y_i into $z_i \in \mathbb{R}^d$ with "small" d .

Local PCA Manifold Learning
(Zhan et al., 2008);

Clé : "Traversal Sequence of
Local Neighborhoods"



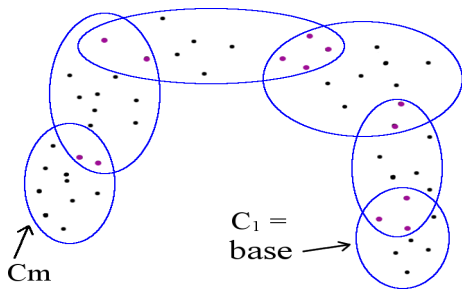
- 1 Series $C_1, \dots, C_m \subset \{1, \dots, n\}$ covering $\{1, \dots, n\}$, with non-empty intersections $C_i \cap C_j, \forall i > j$;
- 2 local PCA on each $C_j \rightarrow$ local z'_j coordinates;

Overview

Goal : transform y_i into $z_i \in \mathbb{R}^d$ with "small" d .

Local PCA Manifold Learning
(Zhan et al., 2008);

Clé : "Traversal Sequence of
Local Neighborhoods"



- 1 Series $C_1, \dots, C_m \subset \{1, \dots, n\}$ covering $\{1, \dots, n\}$, with non-empty intersections $\bigcup_{j < i} C_i \cap C_j, \forall i > 1$;
- 2 local PCA on each $C_j \rightarrow$ local z'_i coordinates;
- 3 Affine transformations \rightarrow global z_i coordinates;
method : optimize transformation matrix on the intersection, then apply on next "cell".

\hookrightarrow

Estimation of global coordinates (steps 2–3)

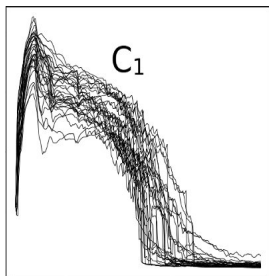


FIG.: data

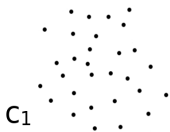


FIG.: local coord.

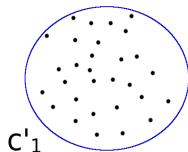


FIG.: global coord.

- 1 PCA on $C_1 \rightarrow$ local coord. c_1 ;
set $c'_1 = c_1$ (global coord.) ;

Estimation of global coordinates (steps 2–3)

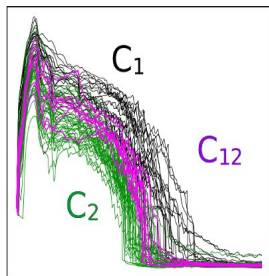


FIG.: data

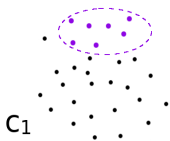


FIG.: local coord.

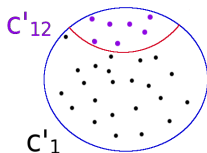


FIG.: global coord.

- 1 PCA on $C_1 \rightarrow$ local coord. c_1 ;
set $c'_1 = c_1$ (global coord.);
- 2 c'_1 contains global coord. of $C_1 \cap C_2$, written c'_{12} ;

Estimation of global coordinates (steps 2–3)

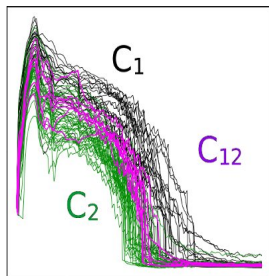


FIG.: data

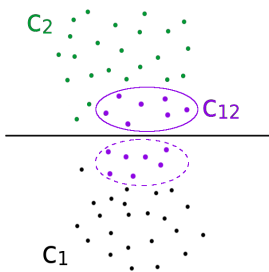


FIG.: local coord.

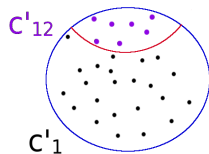


FIG.: global coord.

- 1 PCA on $C_1 \rightarrow$ local coord. c_1 ;
set $c'_1 = c_1$ (global coord.);
- 2 c'_1 contains global coord. of $C_1 \cap C_2$, written c'_{12} ;
- 3 PCA on $C_2 \rightarrow$ local coord. c_2 ;

Estimation of global coordinates (steps 2–3)

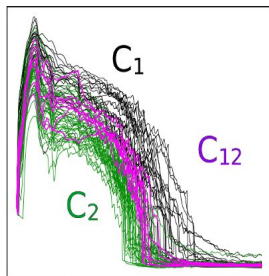


FIG.: data

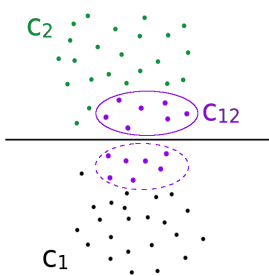


FIG.: local coord.

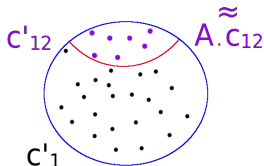


FIG.: global coord.

- 1 PCA on $C_1 \rightarrow$ local coord. c_1 ;
set $c'_1 = c_1$ (global coord.) ;
- 2 c'_1 contains global coord. of $C_1 \cap C_2$, written c'_{12} ;
- 3 PCA on $C_2 \rightarrow$ local coord. c_2 ;
- 4 search for an affine transformation A so that $A c_{12} \simeq c'_{12}$;

Estimation of global coordinates (steps 2–3)

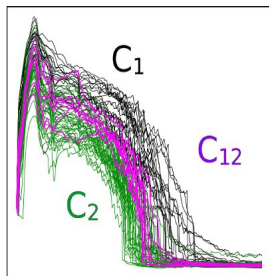


FIG.: data

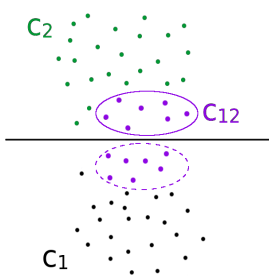


FIG.: local coord.

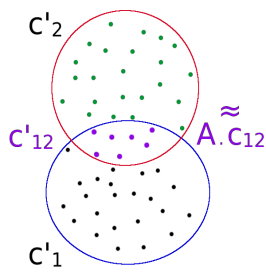


FIG.: global coord.

- 1 PCA on $C_1 \rightarrow$ local coord. c_1 ;
set $c'_1 = c_1$ (global coord.);
- 2 c'_1 contains global coord. of $C_1 \cap C_2$, written c'_{12} ;
- 3 PCA on $C_2 \rightarrow$ local coord. c_2 ;
- 4 search for an affine transformation A so that $A c_{12} \simeq c'_{12}$;
- 5 A applied on the elements of $C_2 \setminus C_1 \rightarrow c'_2$;
back to 2. with $c'_2 \dots$ etc.

Exemples

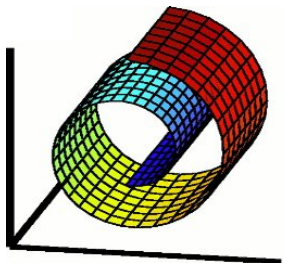


FIG.: Swissroll, 400 points 3D

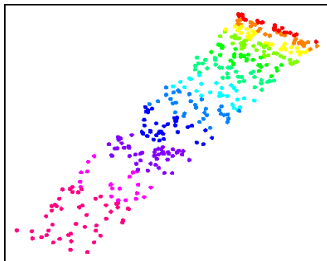


FIG.: LPcaML representation

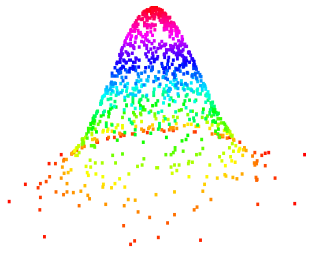


FIG.: Gaussian, 1000 points 3D

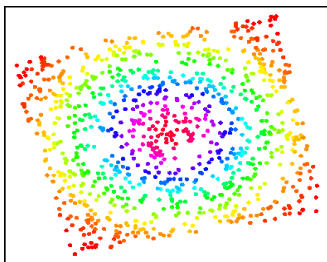


FIG.: LPcaML Representation

1 "Stepwise" dimension reduction

- Local PCA Manifold Learning (Zhan et al. 2008)
- Riemannian Manifold Learning (Lin et al. 2006)

2 Applications

Local steps

RML \simeq preservation of angles *and* geodesic distances.

- 1 choose an origin curve y_0 among the y_i , (e.g., the mean);

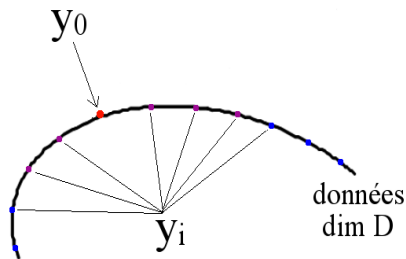


FIG.: Origin curve y_0

Local steps

RML \simeq preservation of angles *and* geodesic distances.

- 1 choose an origin curve y_0 among the y_i , (e.g., the mean);
- 2 determine a local basis $Q_0 = (e_1, \dots, e_d)$ for the tangent space at y_0 (PCA on the neighborhoods curves);

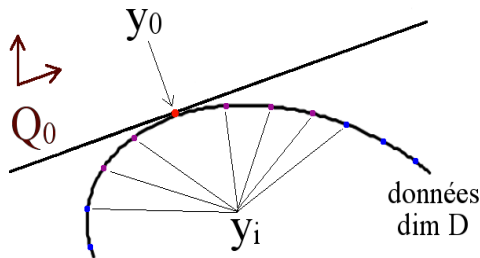


FIG.: Tangent plane at y_0 + local basis Q_0

Local steps

RML \simeq preservation of angles *and* geodesic distances.

- 1 choose an origin curve y_0 among the y_i , (e.g., the mean);
- 2 determine a local basis $Q_0 = (e_1, \dots, e_d)$ for the tangent space at y_0 (PCA on the neighborhoods curves);
- 3 compute the reduced coordinates z_i or curves y_i "close" to y_0 by projecting on Q_0 ,

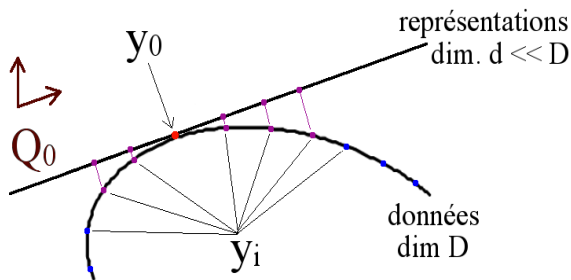


FIG.: Local coordinates z_i on the tangent plane

Local steps

RML \simeq preservation of angles *and* geodesic distances.

- 1 choose an origin curve y_0 among the y_i , (e.g., the mean);
- 2 determine a local basis $Q_0 = (e_1, \dots, e_d)$ for the tangent space at y_0 (PCA on the neighborhoods curves);
- 3 compute the reduced coordinates z_i or curves y_i "close" to y_0 by projecting on Q_0 ,
- 4 then normalize to verify the identity $\|y - y_0\| = \|z - z_0\|$.

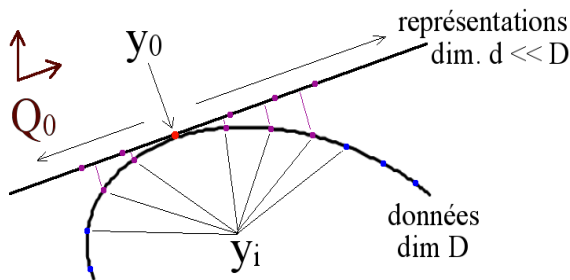


FIG.: Normalization of coordinates z_i

"far" from y_0

Step 4 : for y far from y_0 (too far for last step to be accurate),

- y_p = predecessor of y on a shortest path from y_0
- $y_{i_1}, \dots, y_{i_d} = y_p$ neighbors with known z_i coordinates

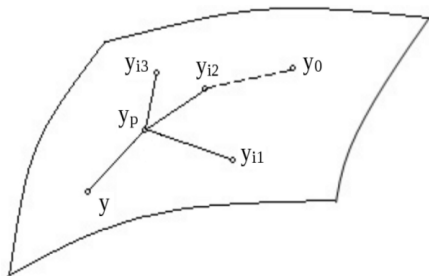


FIG.: Data y_i in dim. D

"far" from y_0

Step 4 : for y far from y_0 (too far for last step to be accurate),

- y_p = predecessor of y on a shortest path from y_0
- $y_{i_1}, \dots, y_{i_d} = y_p$ neighbors with known z_i coordinates

$z = r(y)$ computed by..

- ..preserving angles as much as possible : $\widehat{z z_p z_{i_j}} \simeq \widehat{y y_p y_{i_j}}$;
- ..under the normalization constraint $\|y - y_p\| = \|z - z_p\|$.

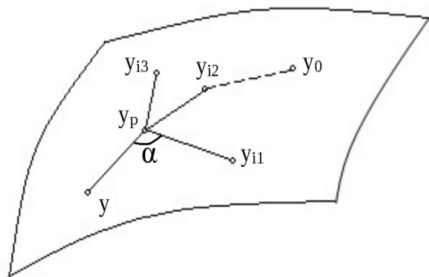


FIG.: Data y_i in dim. D

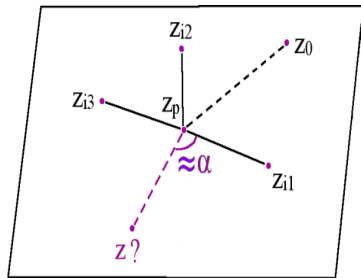


FIG.: $z_i = r(y_i)$ in dim. $d \ll D$

Examples

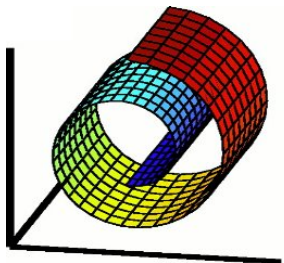


FIG.: Swissroll, 400 points 3D

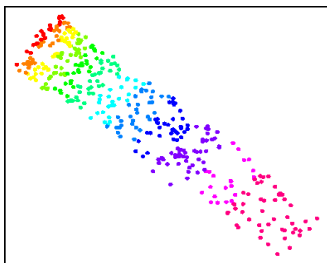


FIG.: RML representation

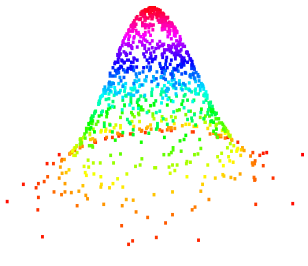


FIG.: Gaussian, 1000 points 3D

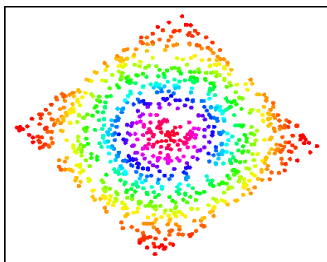


FIG.: RML Representation

About reconstruction...

Assume $z \in \mathbb{R}^d$ must be turned into a curve (\mathbb{R}^D).

RML :

- 1 local PCA around y_{i_0} where i_0 is $\arg \min_{i=1..n} \|z - z_i\|$;
- 2 solve with reversed roles ($y \leftarrow z, z \leftarrow \text{loc. PCA coord.}$)

About reconstruction...

Assume $z \in \mathbb{R}^d$ must be turned into a curve (\mathbb{R}^D).

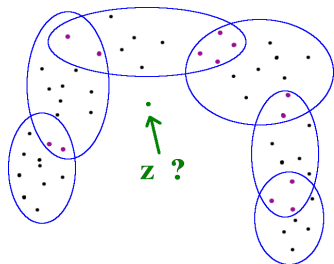
RML :

- 1 local PCA around y_{i_0} where i_0 is $\arg \min_{i=1..n} \|z - z_i\|$;
- 2 solve with reversed roles ($y \leftarrow z$, $z \leftarrow$ loc. PCA coord.)

LPcaML :

- 1 find the most probable cell(s) from which z was generated ;
- 2 inverse local affine transformation \Rightarrow PCA coord., then curve.

..Step 1 = supervised classification ! (Training set = (z_i, cell_i)).



Which cell(s)/weight(s) to choose ?

About reconstruction...

Assume $z \in \mathbb{R}^d$ must be turned into a curve (\mathbb{R}^D).

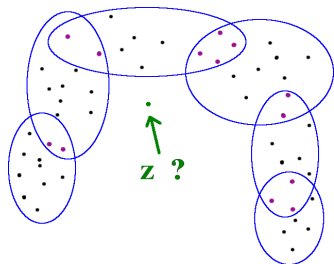
RML :

- 1 local PCA around y_{i_0} where i_0 is $\arg \min_{i=1..n} \|z - z_i\|$;
- 2 solve with reversed roles ($y \leftarrow z$, $z \leftarrow$ loc. PCA coord.)

LPcaML :

- 1 find the most probable cell(s) from which z was generated ;
- 2 inverse local affine transformation \Rightarrow PCA coord., then curve.

..Step 1 = supervised classification ! (Training set = (z_i, cell_i)).



Which cell(s)/weight(s) to choose?

Implemented : weighted average of neighbors' cells reconstructions.

Doesn't work well ...

1 "Stepwise" dimension reduction

- Local PCA Manifold Learning (Zhan et al. 2008)
- Riemannian Manifold Learning (Lin et al. 2006)

2 Applications

Validation step

Data :

- training = $\{(x_i, y_i), i = 1, \dots, n\}$;
- test = $\{(x'_i, y'_i), i = 1, \dots, m\}$;

Model predictions : $\hat{y}'_i = M(x'_i), i = 1, \dots, m.$

Validation step

Data :

- training = $\{(x_i, y_i), i = 1, \dots, n\}$;
- test = $\{(x'_i, y'_i), i = 1, \dots, m\}$;

Model predictions : $\hat{y}'_i = M(x'_i), i = 1, \dots, m$.

"Absolute" then relative measures of the pointwise error

$$MSE[j] = \frac{1}{m} \sum_{i=1}^m (\hat{y}'_i(j) - y'_i(j))^2, \quad j = 1, \dots, D \text{ (discretization).}$$

Validation step

Data :

- training = $\{(x_i, y_i), i = 1, \dots, n\}$;
- test = $\{(x'_i, y'_i), i = 1, \dots, m\}$;

Model predictions : $\hat{y}'_i = M(x'_i), i = 1, \dots, m$.

”Absolute” then relative measures of the pointwise error

$$MSE[j] = \frac{1}{m} \sum_{i=1}^m (\hat{y}'_i(j) - y'_i(j))^2, \quad j = 1, \dots, D \text{ (discretization).}$$

$$Q_2[j] = 1 - \frac{m \cdot MSE[j]}{\sum_{i=1}^m (\bar{y}(j) - y'_i(j))^2} \text{ (compararison to the mean).}$$

$-\infty < Q_2 \leq 1$: $\leq 0 \Rightarrow$ (very) bad model ;
 $\simeq 1 \Rightarrow$ perfect model.

Test I - temperature curves

100 model runs,
4 dimensions on input,
168 discretization points.

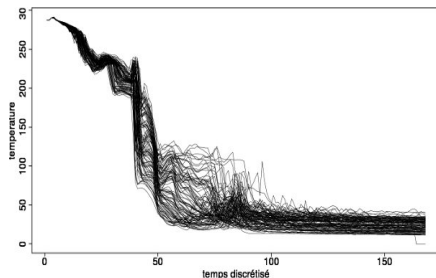


FIG.: The 100 code outputs

Test I - temperature curves

100 model runs,
4 dimensions on input,
168 discretization points.

cross validation
leave-10-out :

MSE at l., Q_2 at r. ; $d = 4$

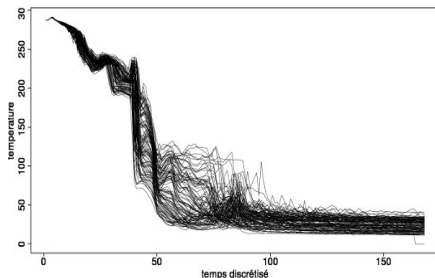


FIG.: The 100 code outputs

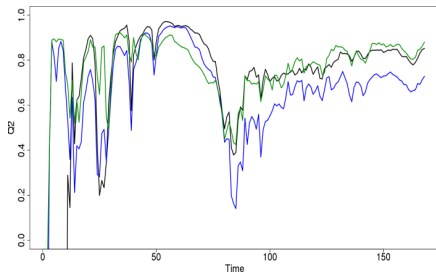
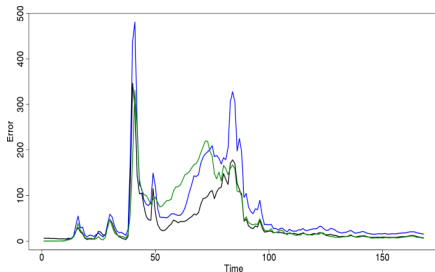


FIG.: Black : functional PCA ; blue : RML ; green : Nadaraya-Watson.

5 predicted curves

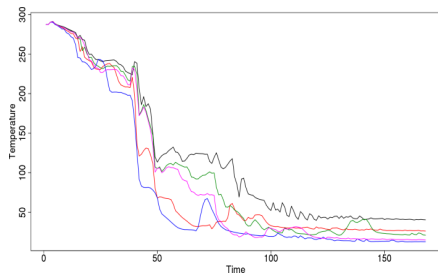


FIG.: Real curves

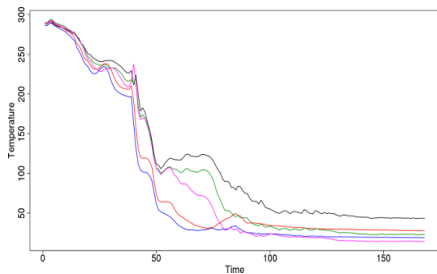


FIG.: PCA dim. red.

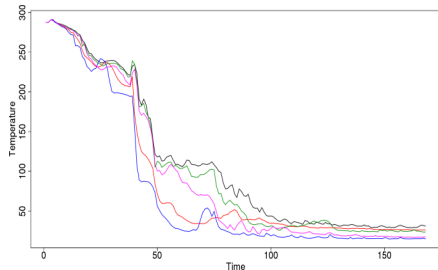


FIG.: RML dim. red.

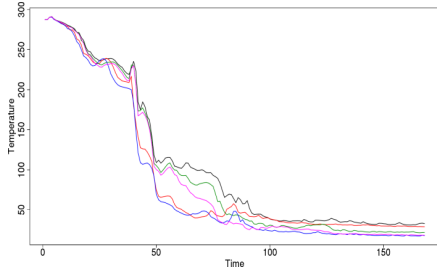


FIG.: Nadaraya-Watson

Test II - temperature curves

600 model runs,
11 dimensions on input,
414 discretization points.

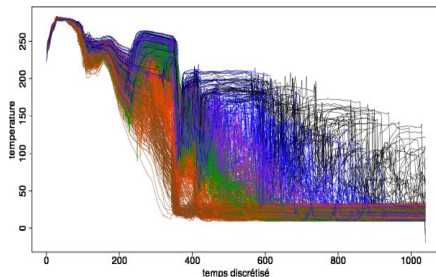


FIG.: The 600 code outputs

Test II - temperature curves

600 model runs,
11 dimensions on input,
414 discretization points.

cross validation
leave-10-out :

MSE at l., Q_2 at r.; $d = 7$

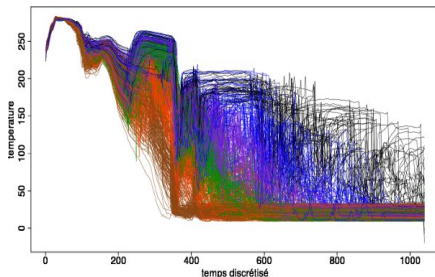


FIG.: The 600 code outputs

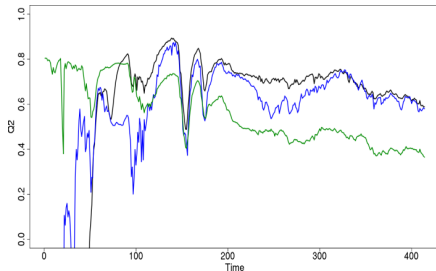
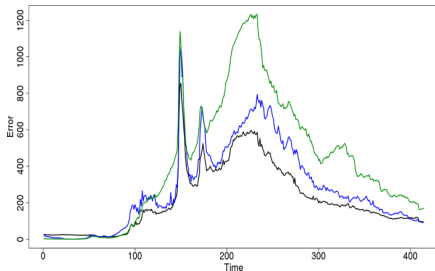


FIG.: Black : functional PCA ; blue : RML ; green : Nadaraya-Watson.

5 predicted curves

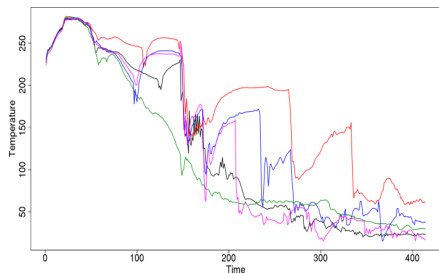


FIG.: Real curves

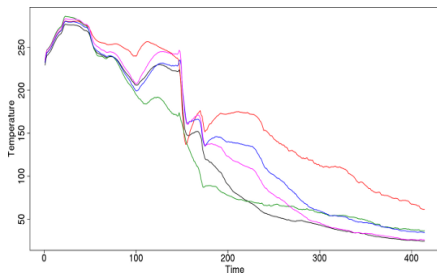


FIG.: PCA red. dim.

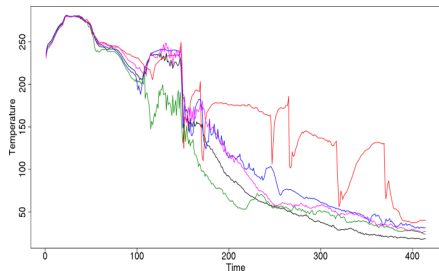


FIG.: RML red. dim.

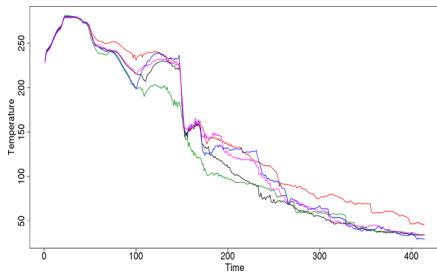


FIG.: Nadaraya-Watson

Conclusion

The model is good enough regarding to the industrial context.

⇒ help to the projet DDVCV (life span of reactor vessels).

Conclusion

The model is good enough regarding to the industrial context.

⇒ help to the projet DDVCV (life span of reactor vessels).

Dimensionality reduction

Linear (PCA) and non-linear : complementary.

Conclusion

The model is good enough regarding to the industrial context.

⇒ help to the projet DDVCV (life span of reactor vessels).

Dimensionality reduction

Linear (PCA) and non-linear : complementary.

Missing : proof of convergence to the right manifold ...

...and some parameters which should be "optimized" automatically.

Conclusion

The model is good enough regarding to the industrial context.

⇒ help to the projet DDVCV (life span of reactor vessels).

Dimensionality reduction

Linear (PCA) and non-linear : complementary.

Missing : proof of convergence to the right manifold ...

...and some parameters which should be "optimized" automatically.

Future research : "functional" principal curves and surfaces.

Example of a principal surface in $2D$:

