

# Construction d'un métamodèle à sorties fonctionnelles

Benjamin Auder

CEA - UPMC

6 octobre 2009

Thèse depuis 02/2008

Directeur de thèse : Gérard Biau (UPMC)

Encadrant CEA : Bertrand Iooss (CEA)

# Contexte industriel global

Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées ...

*But* : majorer la probabilité de chaque accident répertorié.

# Contexte industriel global

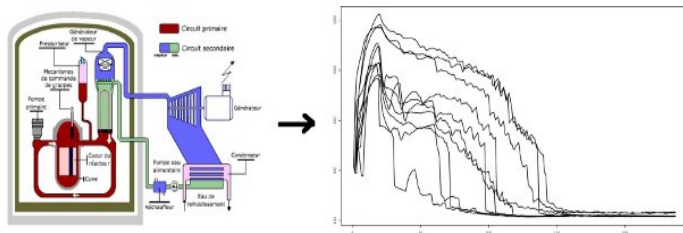
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées ...

*But* : majorer la probabilité de chaque accident répertorié.

## Méthodologie

Modélisation → Simulation → Calculs.



→ Analyse de sensibilité, propagation d'incertitudes ..etc.

# Contexte industriel global

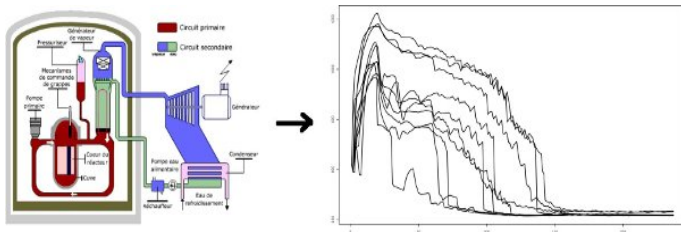
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées ...

*But* : majorer la probabilité de chaque accident répertorié.

## Méthodologie

Modélisation → Simulation → Calculs.



→ Analyse de sensibilité, propagation d'incertitudes ..etc.

*Améliorer la phase simulation, pour effectuer des calculs plus fiables.*

# Motivations au DER/SESI/LCFR

## Cathare (CEA)

Code thermo-hydraulique **coûteux** en temps, déterminant les évolutions temporelles de paramètres physiques dans l'espace annulaire de la cuve.

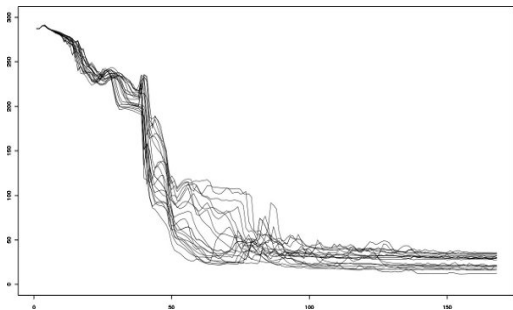


FIG.: Transitoires de température.

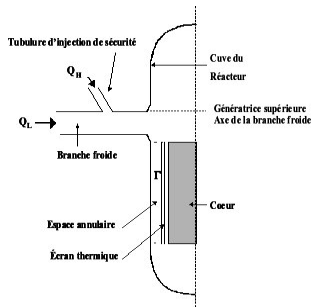


FIG.: Zone modélisée

# Résultats attendus, étapes suivies

$N$  couples  $(x_i, y_i)$  connus :

- Entrées  $x_i \in \mathbb{R}^p =$  état initial du système physique ;
- Sorties  $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$  évolution des paramètres physiques.

*Objectif* : prédiction de données fonctionnelles via un **métamodèle** :

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

# Résultats attendus, étapes suivies

$N$  couples  $(x_i, y_i)$  connus :

- Entrées  $x_i \in \mathbb{R}^p =$  état initial du système physique ;
- Sorties  $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$  évolution des paramètres physiques.

*Objectif* : prédiction de données fonctionnelles via un **métamodèle** :

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

## Hypothèse

Les sorties peuvent être divisées en  $k$  groupes.

⇒ Il faut **retrouver ces groupes**.

# Résultats attendus, étapes suivies

$N$  couples  $(x_i, y_i)$  connus :

- Entrées  $x_i \in \mathbb{R}^p =$  état initial du système physique ;
- Sorties  $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$  évolution des paramètres physiques.

*Objectif* : prédiction de données fonctionnelles via un **métamodèle** :

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

## Hypothèse

Les sorties peuvent être divisées en  $k$  groupes.

⇒ Il faut **retrouver ces groupes**.

Cas "facile" :  $y_i \in \mathbb{R}^d$

⇒ **Représenter une fonction** avec  $d$  réels..

Puis apprentissage statistique classique.



# Résultats attendus, étapes suivies

$N$  couples  $(x_i, y_i)$  connus :

- Entrées  $x_i \in \mathbb{R}^p =$  état initial du système physique ;
- Sorties  $y_i \in \mathcal{C}([a, b], \mathbb{R}) =$  évolution des paramètres physiques.

*Objectif* : prédiction de données fonctionnelles via un **métamodèle** :

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

## Hypothèse

Les sorties peuvent être divisées en  $k$  groupes.

⇒ Il faut **retrouver ces groupes**.

Cas "facile" :  $y_i \in \mathbb{R}^d$

⇒ **Représenter une fonction** avec  $d$  réels..

Puis apprentissage statistique classique.

..Puis *reconstruction* !

## 1 Clustering

- Motivations et difficultés
- Distance " Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemaniann Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

## 1 Clustering

- Motivations et difficultés
- Distance " Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Exemple : 100 transitoires de température

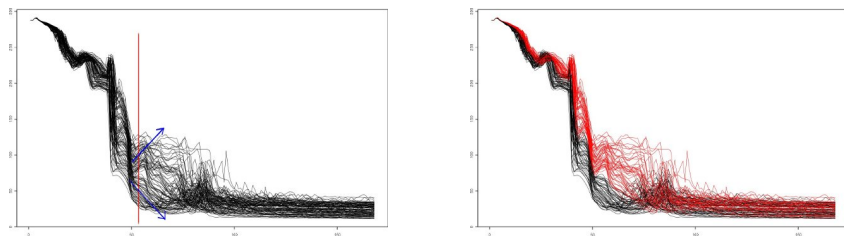


FIG.: Deux groupes de courbes ?

# Exemple : 100 transitoires de température

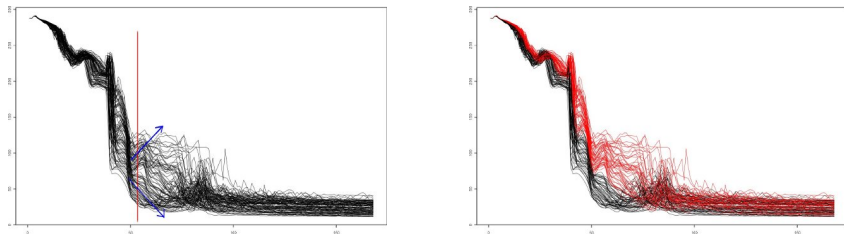


FIG.: Deux groupes de courbes ?

→ Regroupement des courbes aux caractéristiques similaires :

- différents types de comportements physiques ;
- meilleure modélisation dans chaque cluster.

# Précautions à prendre

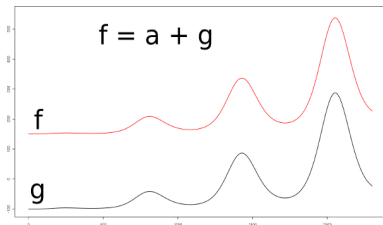
- Pas de clusters trop petits.

# Précautions à prendre

- Pas de clusters trop petits.
- Clusters sur les entrées  $x_i$  ? Nombre de clusters ?  
Possibilité : validation croisée . . .

# Précautions à prendre

- Pas de clusters trop petits.
- Clusters sur les entrées  $x_i$ ? Nombre de clusters?  
Possibilité : validation croisée ...

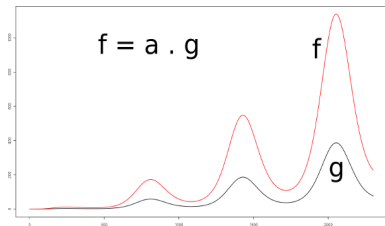
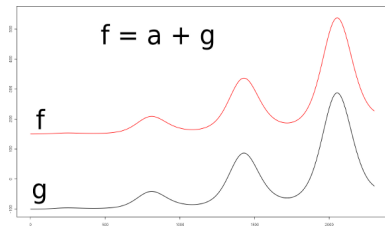


- Type de distance / similarité?
  - ▶ distances  $L^p$  : pas d'information sur les similarités de forme ;



# Précautions à prendre

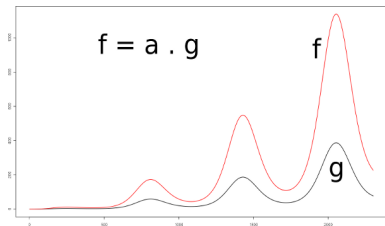
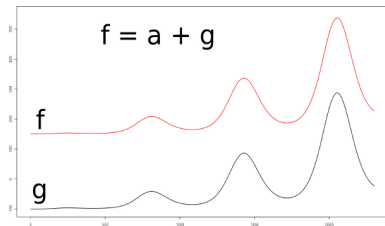
- Pas de clusters trop petits.
- Clusters sur les entrées  $x_i$ ? Nombre de clusters?  
Possibilité : validation croisée ...



- Type de distance / similarité?
  - ▶ distances  $L^p$  : pas d'information sur les similarités de forme ;
  - ▶ semi-distance  $d(f, g) = \int_a^b |f'(t) - g'(t)| dt$  ;

# Précautions à prendre

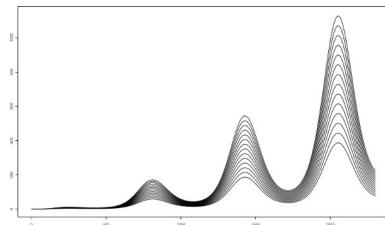
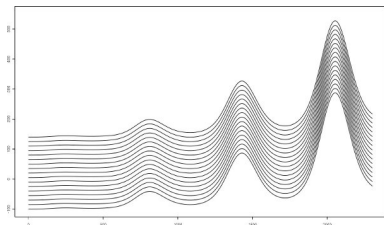
- Pas de clusters trop petits.
- Clusters sur les entrées  $x_i$ ? Nombre de clusters?  
Possibilité : validation croisée ...



- Type de distance / similarité ?
  - ▶ distances  $L^p$  : pas d'information sur les similarités de forme ;
  - ▶ semi-distance  $d(f, g) = \int_a^b |f'(t) - g'(t)| dt$  ;
  - ▶ semi-distance  $d(f, g) = \min_{\alpha, \beta \in \mathbb{R}} \|(\alpha f + \beta \mathbf{1}) + g\|$ .

# Précautions à prendre

- Pas de clusters trop petits.
- Clusters sur les entrées  $x_i$ ? Nombre de clusters?  
Possibilité : validation croisée ...



- Type de distance / similarité?
  - ▶ distances  $L^p$  : pas d'information sur les similarités de forme ;
  - ▶ semi-distance  $d(f, g) = \int_a^b |f'(t) - g'(t)| dt$  ;
  - ▶ semi-distance  $d(f, g) = \min_{\alpha, \beta \in \mathbb{R}} \|(\alpha f + \beta \mathbf{1}) + g\|$ .

## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Graphe de voisinage

Pourquoi ne pas utiliser la distance euclidienne ?

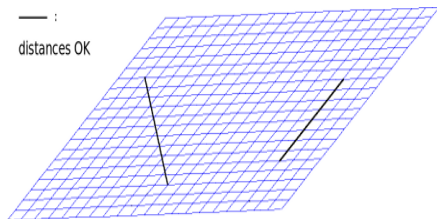


FIG.: Sous-espace linéaire

# Graphe de voisinage

Pourquoi ne pas utiliser la distance euclidienne ?

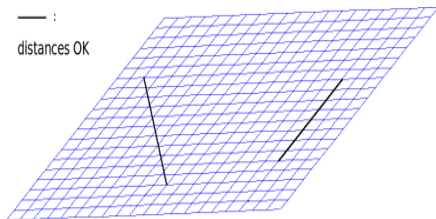


FIG.: Sous-espace linéaire

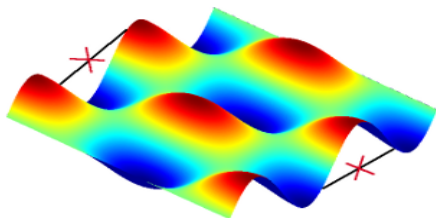


FIG.: Surface plus compliquée..

→ **Prise en compte des déformations de la surface.**

# Graphe de voisinage

Pourquoi ne pas utiliser la distance euclidienne ?

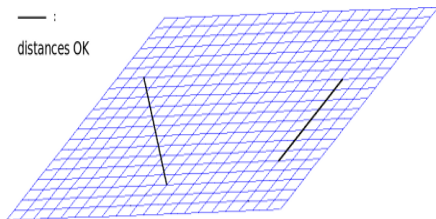


FIG.: Sous-espace linéaire

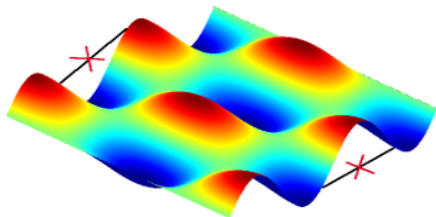
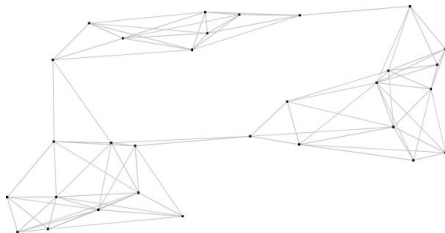


FIG.: Surface plus compliquée..

→ **Prise en compte des déformations de la surface.**

*Approximation : plus courts chemins dans un graphe.*

Exemple de graphe des  $k$  plus proches voisins :



## Exemple : fleuve

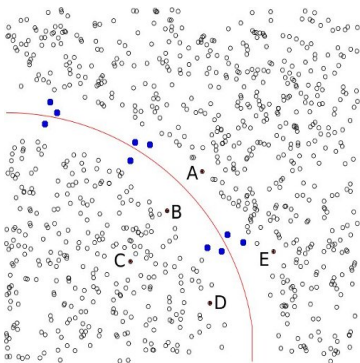


FIG.: Zones d'habitation séparées par un cours d'eau.

*Distance géodésique* :  $|AB| > |BC|$  OK, mais  $|DC| \simeq |DE|$  : faux.

Comment exprimer à la fois  $|AB| > |BC|$  et  $|DC| < |DE|$  ?



# Parcours sur le graphe

"Il y a beaucoup de courts chemins de  $A$  à  $B$ "

= "Une marche aléatoire issue de  $A$  reviendra vite vers  $A$ ".

# Parcours sur le graphe

"Il y a beaucoup de courts chemins de  $A$  à  $B$ "

= "Une marche aléatoire issue de  $A$  reviendra vite vers  $A$ ".

## Marche aléatoire dans le graphe de voisinage

Transition de  $y_i$  vers  $y_j$  avec probabilité

$$\mathbb{P}(i \rightarrow j) = \begin{cases} \alpha e^{-\frac{\|y_i - y_j\|^2}{\sigma_i^2}} & \text{si } y_j \text{ voisin de } y_i, \\ 0 & \text{sinon.} \end{cases}$$

$\sigma_i$  : paramètre local dépendant de  $k$  et ...

# Expression de la distance CT

Principal résultat (M. Saerens et al., 2004 [5])

Temps moyen d'aller-retour entre  $A$  et  $B$

=

carré de la distance euclidienne entre  $A$  et  $B$  dans un espace de représentation à  $N$  dimensions.

# Expression de la distance CT

## Principal résultat (M. Saerens et al., 2004 [5])

Temps moyen d'aller-retour entre  $A$  et  $B$

=

carré de la distance euclidienne entre  $A$  et  $B$  dans un espace de représentation à  $N$  dimensions.

Calcul facile :

- $P$  = matrice de transition,  $P_{ij} = \mathbb{P}(i \rightarrow j)$  ;
- $L$  laplacien du graphe :  $L = I - P$ .

$$d_{CT}^2(y_i, y_j) \propto L_{ii}^+ + L_{jj}^+ - 2L_{ij}^+,$$

avec  $L^+$  pseudo-inverse de  $L$ .

## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Clustering "CT-hiérarchique"

Algorithme déjà conçu pour une matrice de distances.

*Principe* : partant de  $N$  clusters, les fusionner petit à petit en fonction des distances relatives.

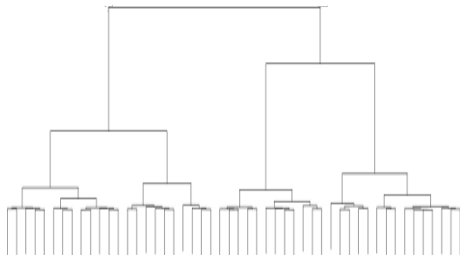


FIG.: Dendrogramme résultat d'un clustering hiérarchique.

# Clustering "CT-hiérarchique"

Algorithme déjà conçu pour une matrice de distances.

*Principe* : partant de  $N$  clusters, les fusionner petit à petit en fonction des distances relatives.

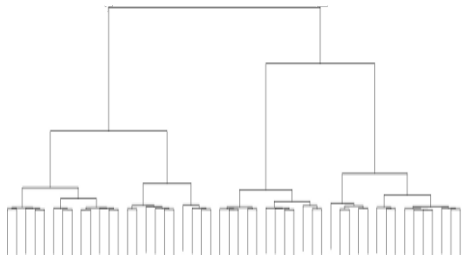


FIG.: Dendrogramme résultat d'un clustering hiérarchique.

Résultat : suite de partitions emboîtées

$(A_i)_{i=1..N}$  avec  $A_{i+1} \prec A_i$ ;  $A_k$  contient  $k$  groupes.

# Exemples

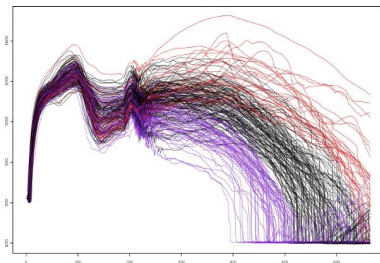
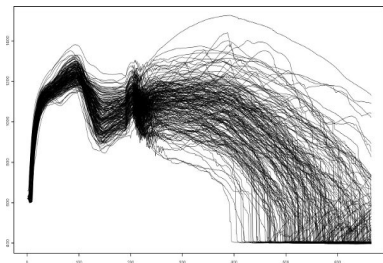


FIG.: 200 sorties du code cathare séparées en 3 groupes

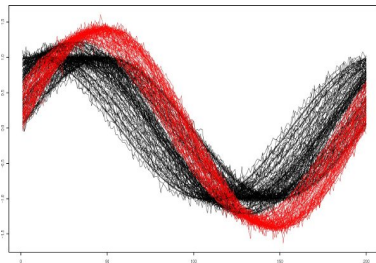
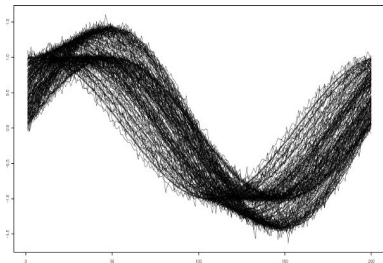


FIG.: 200 courbes dans 2 clusters générés par  $x \mapsto \alpha \cos x + \beta \sin x$



## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Illustration

Données "non linéaires", structurées en une *variété* (au moins)  $\mathcal{C}^0$ .

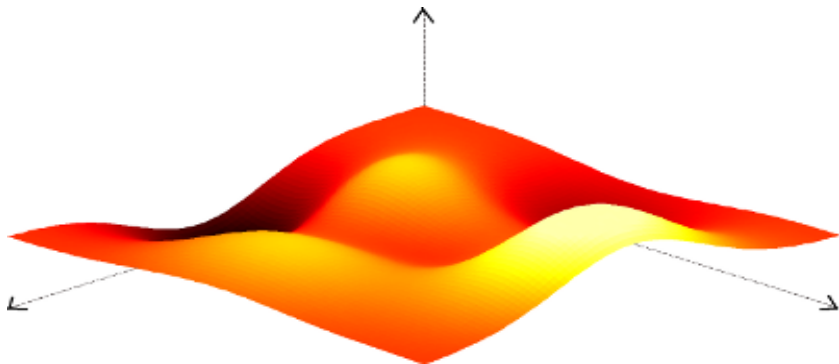


FIG.: Surface de dimension deux dans  $\mathbb{R}^3$ .

*But* : trouver un système de coordonnées le plus réduit possible pour décrire efficacement les données.

## But ultime

Paramétrer  $\mathcal{Y}$ , ensemble des sorties du code :

$f(z \in \mathbb{R}^d) = y \in \mathcal{Y} \subset \mathcal{C}([a, b], \mathbb{R})$ ,  $d$  le plus petit possible.

$N$  échantillons  $y_i \Rightarrow N$  vecteurs  $z_i = f^{-1}(y_i)$  à déterminer.

# But ultime

Paramétrer  $\mathcal{Y}$ , ensemble des sorties du code :

$f(z \in \mathbb{R}^d) = y \in \mathcal{Y} \subset \mathcal{C}([a, b], \mathbb{R})$ ,  $d$  le plus petit possible.

$N$  échantillons  $y_i \Rightarrow N$  vecteurs  $z_i = f^{-1}(y_i)$  à déterminer.

..sous contraintes :

- conservation des voisinages :

les voisins de  $z_i$  correspondent à ceux de  $y_i = f(z_i)$  ( $k \in \mathbb{N}^*$ );

- conservation des distances :

$f(z_i) = y_i$  et  $f(z_j) = y_j \Rightarrow \|z_i - z_j\| \simeq \|y_i - y_j\|$  (..etc)

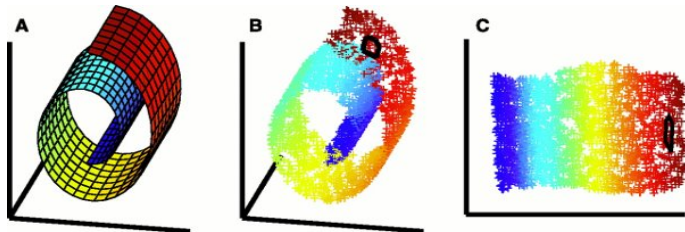


FIG.: carte 2D du jeu de données swissroll

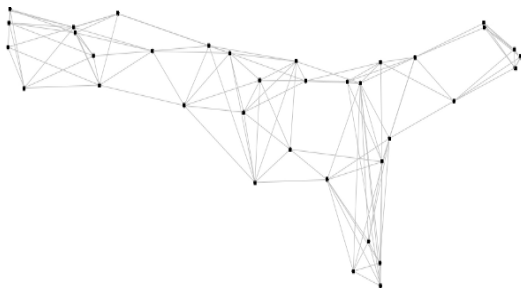
# Méthode

Recherche de la vraie dimension  $\Rightarrow$  représentation non linéaire,  
distances euclidiennes  $\leftarrow$  distances géodésiques.

- 1 Estimation de la géométrie locale : graphe de "visibilité".
- 2 Estimation de la dimension : basée sur  $\mathbb{P}(Y \in B(y, r)) \propto r^d$ .
- 3 Représentation en coordonnées globales selon les contraintes.

Point 1 : ▶ voisinages

Point 2 : ▶ dimension



## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemaniann Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Description (T. Lin & H. Zha, 2006)

Tentative d'unifier des propriétés de distances globales, en respectant aussi les voisinages.

Premières étapes :

- 1 choisir un point origine  $y_0$  parmi les  $y_i$ , (p.ex. le centre) ;
- 2 déterminer une base locale  $Q_0 = (e_1, \dots, e_d)$  de l'espace tangent en  $y_0$  (avec les points du voisinage + SVD) ;

# Description (T. Lin & H. Zha, 2006)

Tentative d'unifier des propriétés de distances globales, en respectant aussi les voisinages.

Premières étapes :

- 1 choisir un point origine  $y_0$  parmi les  $y_i$ , (p.ex. le centre) ;
- 2 déterminer une base locale  $Q_0 = (e_1, \dots, e_d)$  de l'espace tangent en  $y_0$  (avec les points du voisinage + SVD) ;
- 3 calculer les coordonnées de tous les voisins de  $y_0$  en projection sur la base  $Q_0$  ; un voisin  $y$  a pour coordonnées

$$z = \arg \min_{z_1, \dots, z_d} \left\| y - \left( y_0 + \sum_{i=1}^d z_i e_i \right) \right\|^2,$$

renormalisées pour vérifier  $\|y - y_0\| = \|x - x_0\|$ .



## Coordonnées des non voisins de $y_0$

Étape 4 : pour  $y$  non voisin de  $y_0$ , on cherche  $y_p$  le prédécesseur de  $y$  sur un plus court chemin issu de  $y_0$  (Dijkstra p.ex.).  $y_{i_1}, \dots, y_{i_d}$  sont les voisins déjà traités de  $y_p$  (parcours des  $y_i$  en largeur).

→ On cherche alors  $z$  coordonnées de  $y$ , telles que les angles  $\widehat{yy_p y_{i_j}}$  soient  $\simeq$  conservés :

$$\cos \theta = \frac{\langle y - y_p, y_{i_j} - y_p \rangle}{\|y - y_p\| \|y_{i_j} - y_p\|} \simeq \frac{\langle z - z_p, z_{i_j} - z_p \rangle}{\|z - z_p\| \|z_{i_j} - z_p\|} = \cos \theta',$$

sous la contrainte  $\|y - y_p\| = \|z - z_p\|$ .

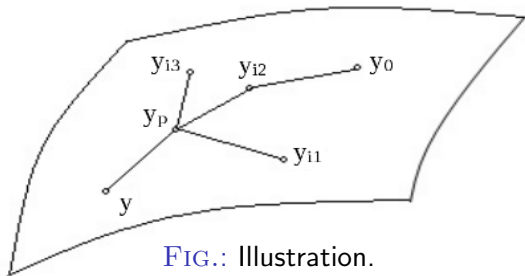


FIG.: Illustration.

# Quelques aspects pratiques

Paramètre utilisateur  $k$  = nombre de voisins en chaque courbe.

Influe sur la forme de la représentation.

*Extension possible* :  $k_i$  local (LTSA ; H. Zha & Z. Zhang, 2003).

⇒ Graphe de visibilité  $\sim$  OK.

# Quelques aspects pratiques

Paramètre utilisateur  $k$  = nombre de voisins en chaque courbe.

Influe sur la forme de la représentation.

*Extension possible* :  $k_i$  local (LTSA ; H. Zha & Z. Zhang, 2003).

⇒ Graphe de visibilité  $\sim$  OK.

Combien de coordonnées réduites en projection sur le plan tangent ?

- Pas assez : imprécision accumulée sur les coordonnées éloignées.
- Trop : chevauchements de zones si replis de la surface.

# Exemples

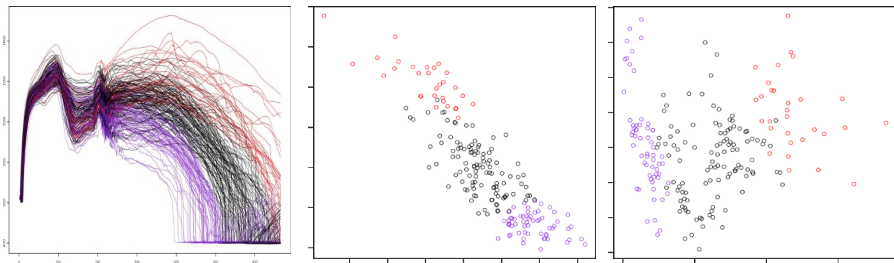


FIG.: g. à d. : 200 sorties du code cathare, coefficients ACP, puis RML

# Exemples

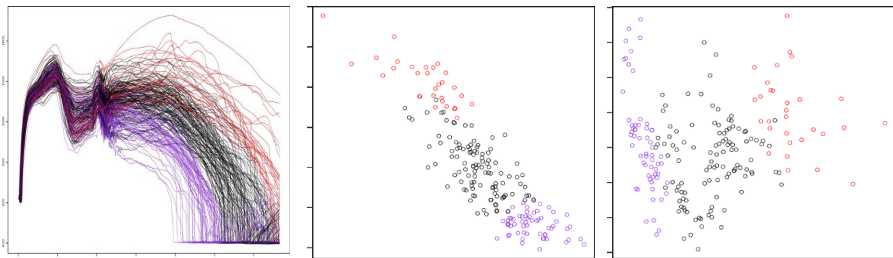


FIG.: g. à d. : 200 sorties du code cathare, coefficients ACP, puis RML

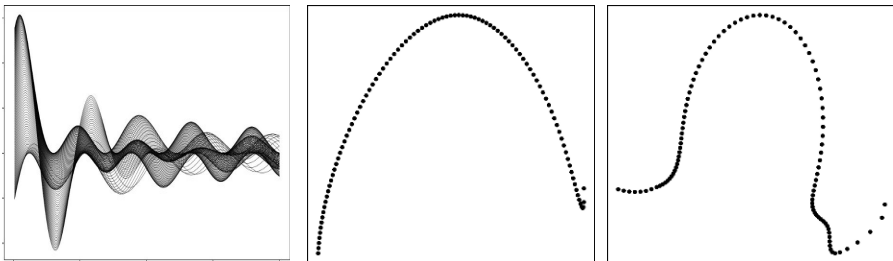


FIG.: Cluster "non linéaire" ; coefficients ACP au centre, RML à droite

## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Exemple analytique, 3 clusters ("dimension 1")

$$(\alpha, \beta, \gamma) \in \mathbb{R}^3 \mapsto f_{\alpha, \beta, \gamma} : [1, 5] \rightarrow \mathbb{R}$$
$$x \mapsto \left( \frac{\sin \alpha x}{x} + e^{-\beta x} \right) \cos \gamma x;$$

Paramètres :  $\alpha_{1,2,3} = 3\beta_{1,2,3}$ ,

$$\beta_1 \sim \mathcal{U}(1, 2), \beta_2 \sim \mathcal{U}(0, 1), \beta_3 \sim \mathcal{U}(0.4, 1),$$

$$\gamma_1 = (4 - \beta_1^2)^{\frac{1}{2}}, \gamma_2 = (1 - \beta_2^2)^{\frac{1}{2}}, \gamma_3 = 3(1 - \beta_3^2)^{\frac{1}{2}} + 3.$$

+ bruit blanc gaussien [▶ test supp.](#)

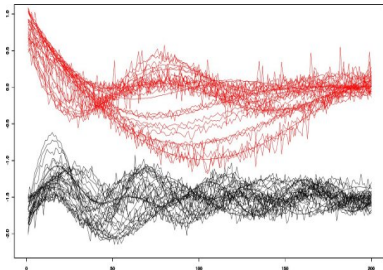


FIG.: 32 courbes des 2 1<sup>ers</sup> clusters.

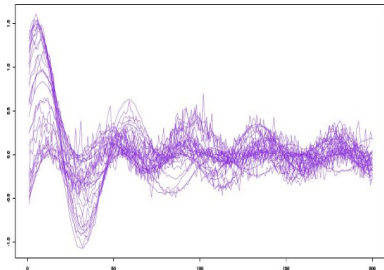


FIG.: 32 courbes du 3<sup>eme</sup> cluster.

# Résultats validation croisée (100-20)

Noir : 150 courbes ; rouge : 300 ; violet : 600.

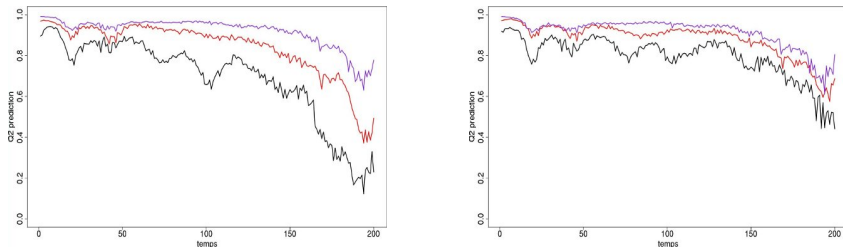


FIG.: Capacité de généralisation ; ACPF à gauche, RML à droite



# Résultats validation croisée (100-20)

Noir : 150 courbes; rouge : 300; violet : 600.

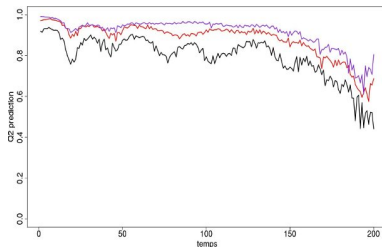
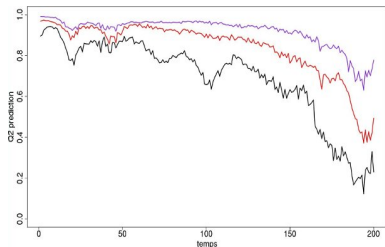


FIG.: Capacité de généralisation; ACPF à gauche, RML à droite

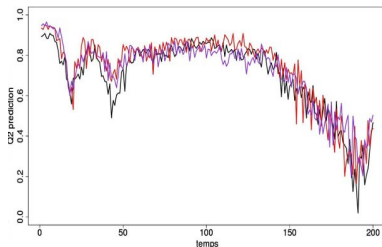
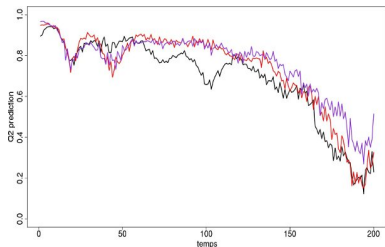


FIG.: ..avec plus de bruit (ACPF à gauche, RML à droite)

## 1 Clustering

- Motivations et difficultés
- Distance "Commute-Time"
- Clustering sur une matrice de distances

## 2 Réduction de la dimension

- Objectifs
- Riemannian Manifold Learning

## 3 Applications

- Fonction dont l'expression est connue
- Jeux de données réels (cathare)

# Transitoires de température ▶ test supp.

100 simulations ;  
4 dimensions en entrée,  
168 points de discrétisation.

entrées-sorties faciles à diviser  
*dimension estimée : 4 (!)*

noir, rouge : PCA +/- clust. ;  
violet, bleu : RML +/- clust.

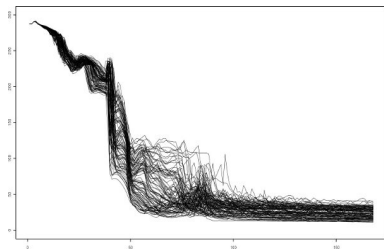


FIG.: Les 100 courbes en sortie

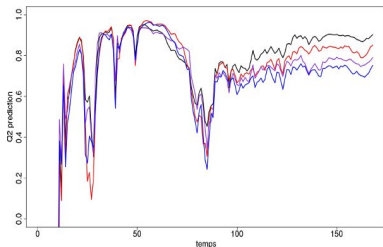


FIG.: Q2 de prédiction (100-10)

# Transitoires de température ▶ test supp.

100 simulations ;  
4 dimensions en entrée,  
168 points de discrétisation.

entrées-sorties faciles à diviser  
*dimension estimée : 4 (!)*

noir, rouge : PCA +/- clust. ;  
violet, bleu : RML +/- clust.

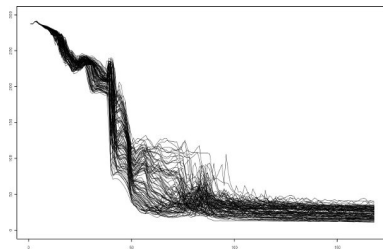


FIG.: Les 100 courbes en sortie

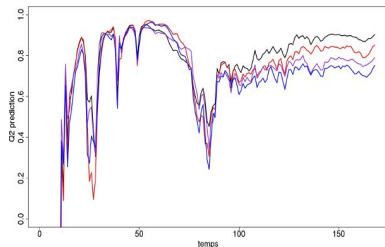


FIG.: Q2 de prédiction (100-10)

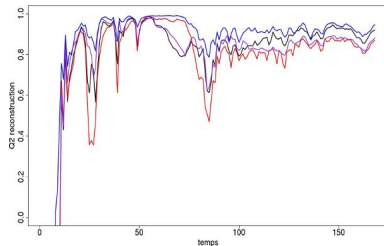


FIG.: Q2 de reconstruction

# Coefficient d'échange fluide-paroi

200 simulations, mais..

44 dimensions en entrée,

665 points de discrétisation.

entrées-sorties difficiles à diviser

*dimension estimée : 5*

noir : PCA sans clustering ;

rouge : RML sans clustering.

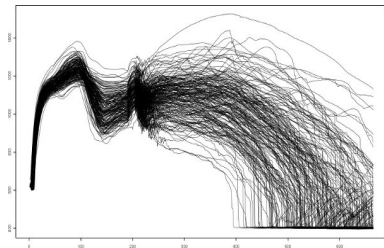


FIG.: Les 200 sorties du code

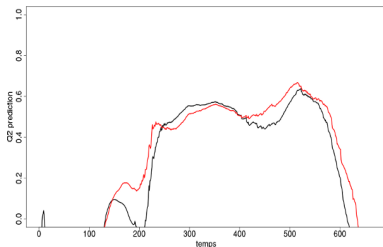


FIG.: Q2 de prédiction (100-20)

# Coefficient d'échange fluide-paroi

200 simulations, mais..

44 dimensions en entrée,

665 points de discrétisation.

entrées-sorties difficiles à diviser

*dimension estimée : 5*

noir : PCA sans clustering ;

rouge : RML sans clustering.

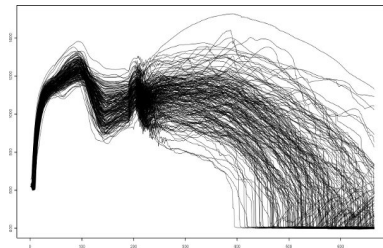


FIG.: Les 200 sorties du code

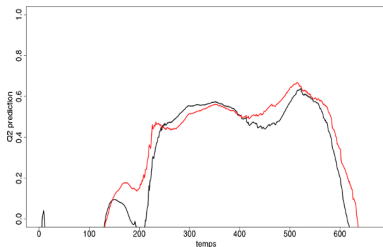


FIG.: Q2 de prédiction (100-20)

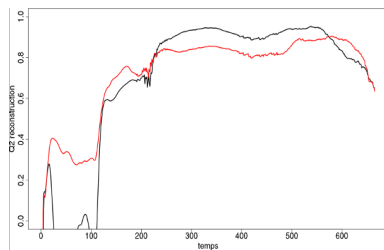


FIG.: Q2 de reconstruction

# Conclusion et perspectives [← plan](#)

Résultats mitigés mais :

- Clustering utile pour certaines données ;
- RML parfois mieux adapté (quand ?).

# Conclusion et perspectives [← plan](#)

Résultats mitigés mais :

- Clustering utile pour certaines données ;
- RML parfois mieux adapté (quand ?).

Algorithme RML + **reconstruction** à améliorer.

encore plusieurs paramètres à optimiser (automatiquement ?).

**Régression** : tenir compte des corrélations inter-coefficients.



# Conclusion et perspectives ◀ plan

Résultats mitigés mais :

- Clustering utile pour certaines données ;
- RML parfois mieux adapté (quand ?).

Algorithme RML + **reconstruction** à améliorer.

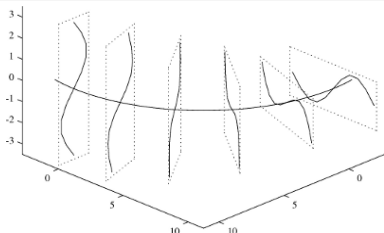
encore plusieurs paramètres à optimiser (automatiquement ?).

**Régression** : tenir compte des corrélations inter-coefficients.

## Méthode alternative à explorer

Courbes, surfaces principales (T. Hastie, 1984 [3]) "fonctionnelles"

Exemple de surface principale en 2D :



 A. M. Farahmand, C. Szepesvári, and J-Y. Audibert.

Manifold-Adaptive Dimension Estimation.

In *24th International Conference on Machine Learning*, pages 265–272, Corvallis, Oregon, USA, 2007.

 J. H. Friedman and W. Stuetzle.

Projection Pursuit Regression.

*Journal of the American Statistical Association*, 76, 1981.

 T. Hastie.

*Principal Curves and Surfaces*.

PhD thesis, Stanford University, 1984.

 T. Lin and H. Zha.

Riemannian Manifold Learning.

*IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30 :796–809, 2008.

 M. Saerens, A. Pirotte, and F. Fouss.

Computing Dissimilarities between Nodes of a Graph.

Technical report, Université catholique de Louvain, 2004.

# Détermination des voisinages

◀ Manifold Learning

Méthode utilisée par T. Lin & H. Zha :

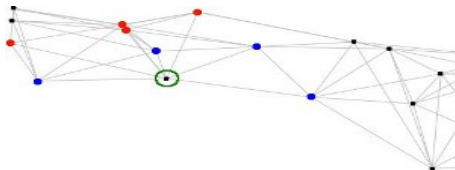
*Riemannian Manifold Learning* [4]

## Définition : visibilité depuis un noeud

$v$  est voisin de  $p$  si aucun autre point  $r$  ne vérifie à la fois

$$\|r - p\| < \|v - p\| \text{ et}$$

$$\langle p - r, v - r \rangle < 0.$$



bleu : points testés et acceptés

rouge : points testés et refusés

FIG.: Exemple : voisinage du sommet entouré en vert.

*Manifold-adaptive dimension estimation*, A. M. Farahmand et al. [1]

## Idée

Pour une distribution uniforme en dimension  $d$ , la probabilité de trouver un élément dans une boule de rayon  $r$  est  $\propto r^d$ .

Supposant la variété à estimer régulière et uniformément échantillonnée :

$\mathbb{P}(Y \in B(y_i, r)) = \eta(y_i, r)r^d$ , avec  $\eta(y_i, \cdot) = \eta_0^{(i)}$  au voisinage de 0.

*Manifold-adaptive dimension estimation*, A. M. Farahmand et al. [1]

## Idée

Pour une distribution uniforme en dimension  $d$ , la probabilité de trouver un élément dans une boule de rayon  $r$  est  $\propto r^d$ .

Supposant la variété à estimer régulière et uniformément échantillonnée :

$\mathbb{P}(Y \in B(y_i, r)) = \eta(y_i, r)r^d$ , avec  $\eta(y_i, \cdot) = \eta_0^{(i)}$  au voisinage de 0.

Si  $y_i$  a  $k_i$  voisins proches avec  $N$  "grand" :  $\mathbb{P}(Y \in B(y_i, r_k^{(i)})) \simeq \frac{k}{N}$ ,  
avec  $r_k^{(i)}$  = distance au  $k^{\text{eme}}$  voisin,  $1 \leq k \leq k_i$ .

## Formule pour $d$

$$\ln \frac{k_i}{N} \simeq \ln \eta_0^{(i)} + d \ln r_{k_i}^{(i)}, \text{ et}$$

$$\ln \frac{k_i}{2N} \simeq \ln \eta_0^{(i)} + d \ln r_{\lceil k_i/2 \rceil}^{(i)}.$$

$$\hat{d}_i = \frac{\ln 2}{\ln r_{k_i}^{(i)} / r_{\lceil k_i/2 \rceil}^{(i)}}$$

= Estimation locale de la dimension, supposant  $k_i$  connu.

Deux options :

- $\hat{d} = \frac{1}{N} \sum_{i=1}^N \hat{d}_i$  ;
- $\hat{d} = \arg \max_{d' \in \mathbb{N}^*} \sum_{i=1}^N \mathbb{1}_{\hat{d}_i = d'}$ .

# Régression entrées - coordonnées réduites

◀ Conclusion

Données :  $(x_1, z_1), \dots, (x_k, z_k)$  dans le cluster  $C$ .  $f(x_0) = ?$

# Régression entrées - coordonnées réduites

◀ Conclusion

Données :  $(x_1, z_1), \dots, (x_k, z_k)$  dans le cluster  $C$ .  $f(x_0) = ?$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, 1981 [2].

Proche d'un perceptron feedforward à une couche cachée.

*Nuance* : interpolation non paramétrique aux noeuds intermédiaires.



Données :  $(x_1, z_1), \dots, (x_k, z_k)$  dans le cluster  $C$ .  $f(x_0) = ?$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, 1981 [2].

Proche d'un perceptron feedforward à une couche cachée.

*Nuance* : interpolation non paramétrique aux noeuds intermédiaires.

## Algorithme

Initialisation  $r_i = z_i, j = 1$ .

- 1 "Projection" : déterminer  $\alpha_j \in \mathbb{R}^p$  direction de projection.

Données :  $(x_1, z_1), \dots, (x_k, z_k)$  dans le cluster  $C$ .  $f(x_0) = ?$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, 1981 [2].

Proche d'un perceptron feedforward à une couche cachée.

*Nuance* : interpolation non paramétrique aux noeuds intermédiaires.

## Algorithme

Initialisation  $r_i = z_i, j = 1$ .

- 1 "Projection" : déterminer  $\alpha_j \in \mathbb{R}^p$  direction de projection.
- 2 "Pursuit regression" : interpolation non paramétrique sur les points  $\langle \alpha_j, x_i \rangle, r_i : m_j(\langle \alpha_j, x_i \rangle) \simeq r_i$ .
- 3  $r_i = r_i - m_j(\langle \alpha_j, x_i \rangle), j = j + 1$  puis étape de projection, jusqu'à obtenir  $q$  termes.

# Reconstitution non linéaire

◀ Conclusion

Données de l'ensemble d'apprentissage :

- la matrice  $C$  des  $N$  coefficients réduits ;
- la matrice  $X$  des  $N$  vecteurs en entrée ;
- la matrice  $Y$  des  $N$  fonctions discrétisées en sortie ;

Pour un vecteur de coefficients réduits  $c_0 = (c_0^{(1)}, \dots, c_0^{(d)})$ , on veut  $\hat{y}_0$  estimation de la courbe correspondante.

# Reconstitution non linéaire

◀ Conclusion

Données de l'ensemble d'apprentissage :

- la matrice  $C$  des  $N$  coefficients réduits ;
- la matrice  $X$  des  $N$  vecteurs en entrée ;
- la matrice  $Y$  des  $N$  fonctions discrétisées en sortie ;

Pour un vecteur de coefficients réduits  $c_0 = (c_0^{(1)}, \dots, c_0^{(d)})$ , on veut  $\hat{y}_0$  estimation de la courbe correspondante.

Méthode :

- 1 recherche du plus proche voisin de  $c_0$  dans  $C$  :  $c_k$  ;
- 2 ACP fonctionnelle locale en  $y_k$  : base  $B$  ;

Données de l'ensemble d'apprentissage :

- la matrice  $C$  des  $N$  coefficients réduits ;
- la matrice  $X$  des  $N$  vecteurs en entrée ;
- la matrice  $Y$  des  $N$  fonctions discrétisées en sortie ;

Pour un vecteur de coefficients réduits  $c_0 = (c_0^{(1)}, \dots, c_0^{(d)})$ , on veut  $\hat{y}_0$  estimation de la courbe correspondante.

Méthode :

- 1 recherche du plus proche voisin de  $c_0$  dans  $C$  :  $c_k$  ;
- 2 ACP fonctionnelle locale en  $y_k$  : base  $B$  ;
- 3 calcul des  $N$  vecteurs de coefficients ACP :  $V$  ;
- 4 algorithme PPR :  $C \rightarrow V$  ;

Données de l'ensemble d'apprentissage :

- la matrice  $C$  des  $N$  coefficients réduits ;
- la matrice  $X$  des  $N$  vecteurs en entrée ;
- la matrice  $Y$  des  $N$  fonctions discrétisées en sortie ;

Pour un vecteur de coefficients réduits  $c_0 = (c_0^{(1)}, \dots, c_0^{(d)})$ , on veut  $\hat{y}_0$  estimation de la courbe correspondante.

Méthode :

- 1 recherche du plus proche voisin de  $c_0$  dans  $C$  :  $c_k$  ;
- 2 ACP fonctionnelle locale en  $y_k$  : base  $B$  ;
- 3 calcul des  $N$  vecteurs de coefficients ACP :  $V$  ;
- 4 algorithme PPR :  $C \rightarrow V$  ;
- 5 estimation de  $v_0$ , puis de  $y_0$ .

# Oscillations amorties ← fconnue

$$f_{\alpha,\beta,\gamma,\delta}(x) = -\alpha^2 e^{-\frac{x^2}{8}} / \sqrt{2\pi} + \beta + e^{\{\text{sgn}(3.5-x)\sqrt{|3.5-x|}\}} \gamma \sin(\delta x)^3$$

$$\alpha \sim \mathcal{U}(1, 5),$$

$$(\beta, \gamma, \delta) \sim \mathcal{U}(S^2(0, 1))$$

noir : 100 courbes ;

rouge : 200 courbes ;

violet : 500 courbes.

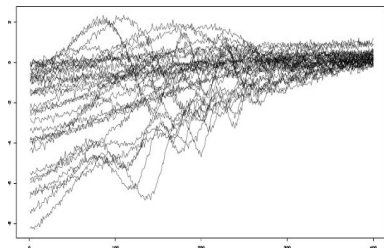


FIG.: 32 courbes au hasard

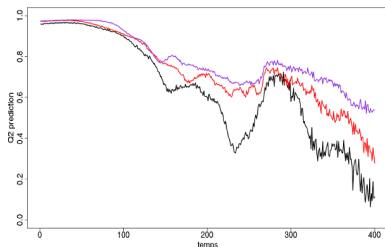
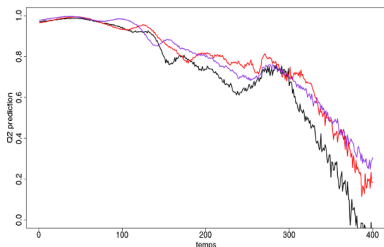


FIG.: Q2 de prédiction (100-20,  $d=3$ ); ACPF à gauche, RML à droite

## Exemple réel : climat ← cathare

Modèle dynamique d'évolution des récoltes journalières (INRA 2008).

Entrée : un vecteur de 7 paramètres + une année (de 1 à 14)

Sortie = variable d'état "biomasse"  $U$ , actualisée jour après jour :

$$U(t + 1) = U(t) + C_1(t) + \varepsilon(t).$$

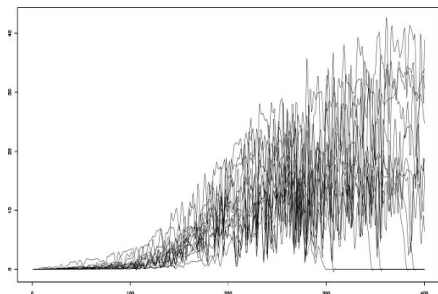


FIG.: Évol. des récoltes sur 14 ans

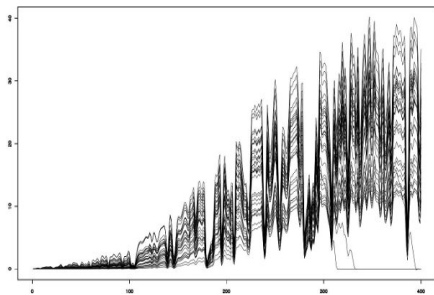


FIG.: Évol. des récoltes, année 7



# Résultats validation croisée (100-20)

entrées-sorties faciles à classer

*dimension estimée : 7*

noir - rouge :

PCA pour 100 - 200 courbes ;

violet - bleu :

RML pour 100 - 200 courbes.

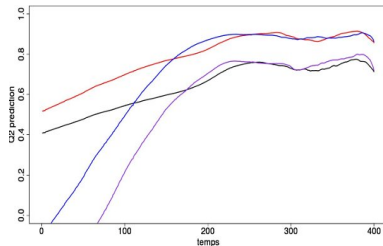


FIG.: Q2 moyenné sur 14 ans

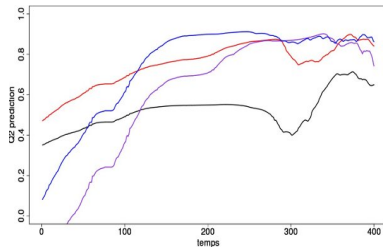
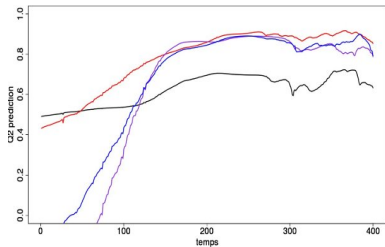


FIG.: Cas RML intéressant ; années 2 à gauche, 7 à droite