

Classification et modélisation de sorties fonctionnelles de codes de calcul.

Benjamin Auder

CEA - UPMC

5 mai 2011

Directeur de thèse (UPMC) : Gérard Biau
Encadrant CEA puis EDF : Bertrand Iooss
Encadrant CEA : Michel Marquès

Contexte industriel

Cadre : durée de vie des cuves.

Contexte industriel

Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

Contexte industriel

Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

But : estimer leurs probabilités.

Contexte industriel

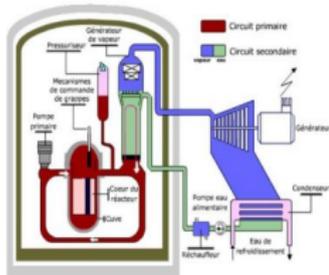
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

But : estimer leurs probabilités.

Méthodologie

Modélisation



Contexte industriel

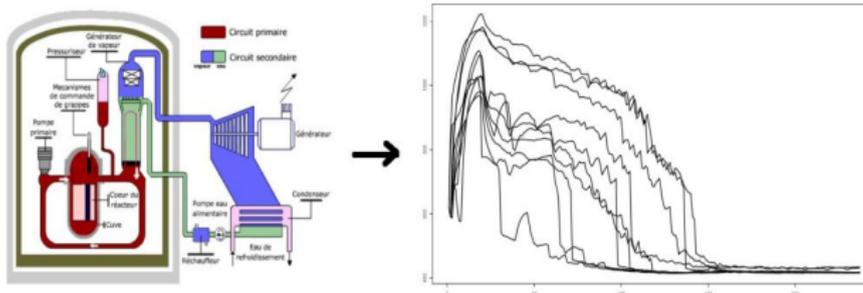
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

But : estimer leurs probabilités.

Méthodologie

Modélisation → Simulation



Contexte industriel

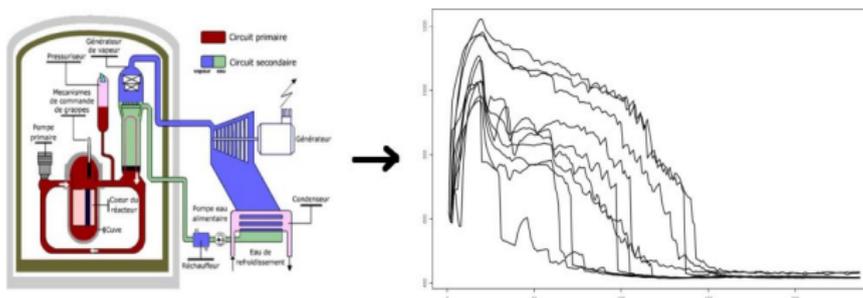
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

But : estimer leurs probabilités.

Méthodologie

Modélisation → Simulation → Calculs.



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

Contexte industriel

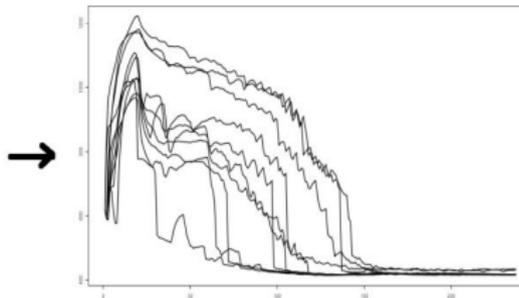
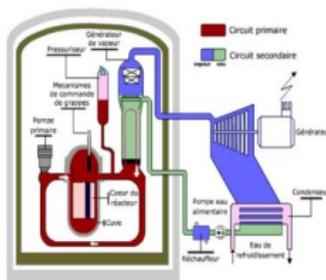
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagés.

But : estimer leurs probabilités.

Méthodologie

Modélisation → **Simulation** → Calculs.



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

Améliorer la phase simulation, pour effectuer des calculs plus fiables

Au CEA - DER/SESI/LSMR ...

Code thermo-hydraulique CATHARE **coûteux** en temps.

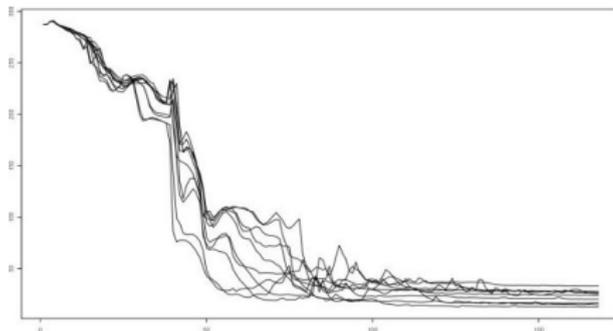


Figure: Transitoires de température.

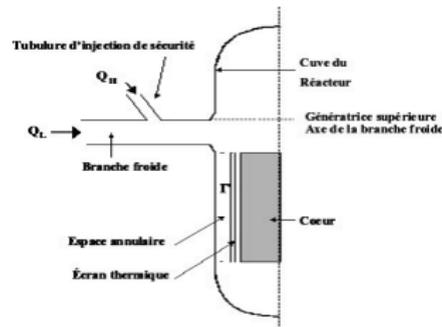


Figure: Zone modélisée

Au CEA - DER/SESI/LSMR ...

Code thermo-hydraulique CATHARE **coûteux** en temps.

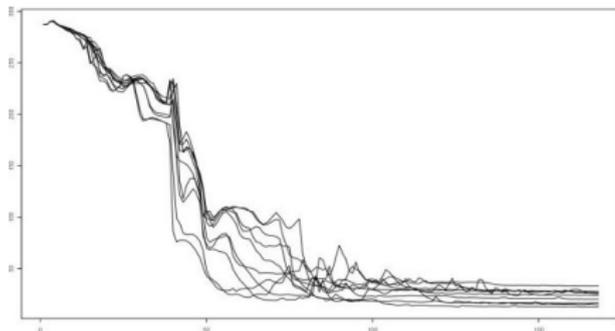


Figure: Transitoires de température.

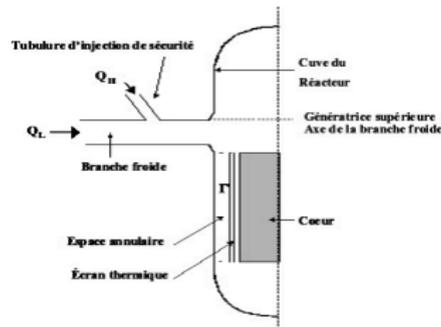


Figure: Zone modélisée

Entrées x_i → **CATHARE** → **CASTEM** → probabilité de défaillance

}
 code thermo-mécanique

Au CEA - DER/SESI/LSMR ...

Code thermo-hydraulique CATHARE **coûteux** en temps.

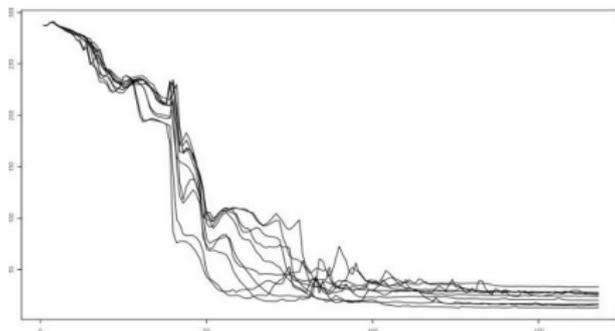


Figure: Transitoires de température.

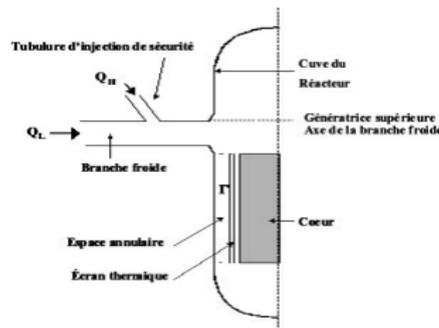
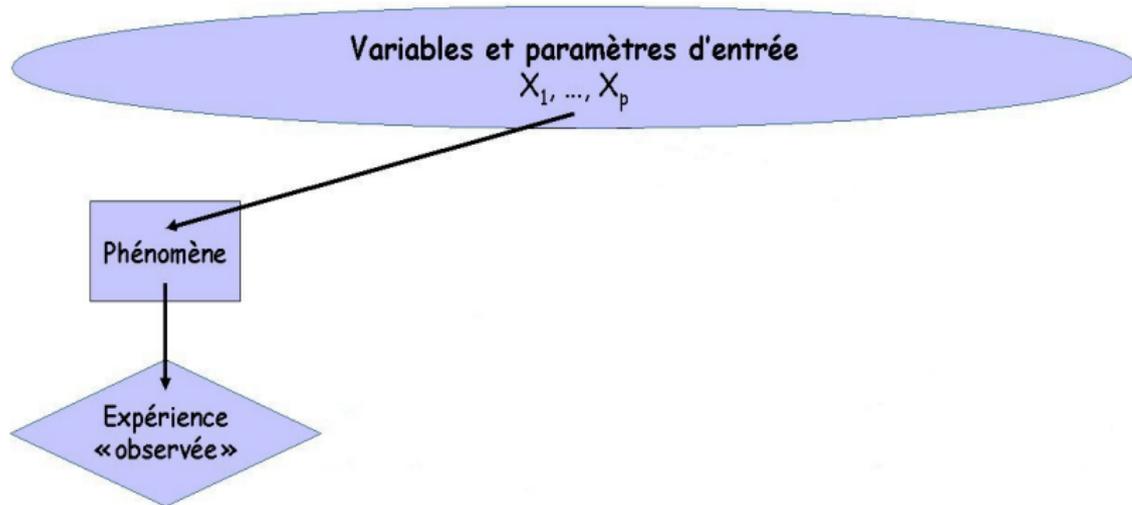


Figure: Zone modélisée

Entrées x_i → **CATHARE** → **CASTEM** → probabilité de défaillance
}
 code thermo-mécanique

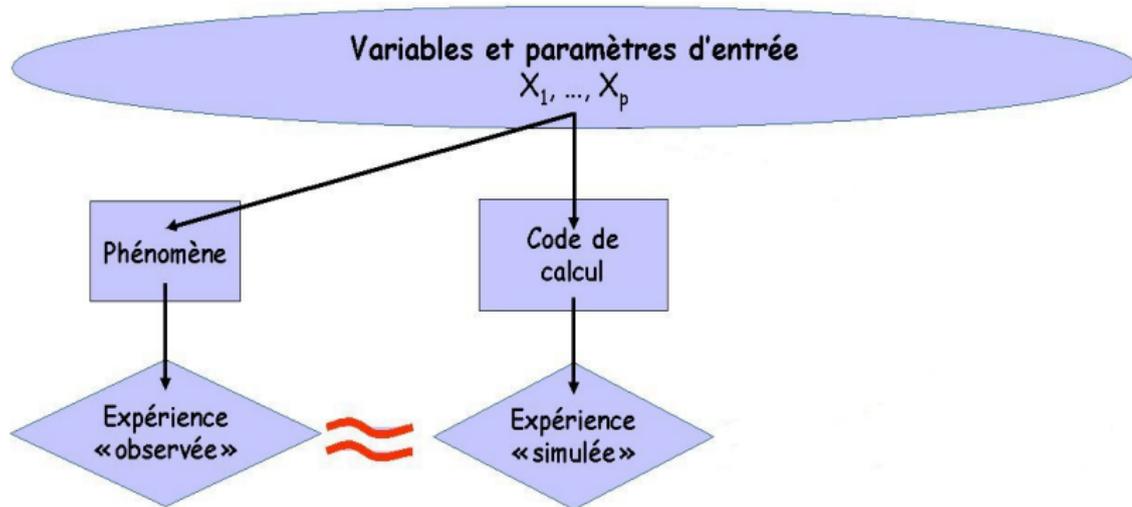
"Accélérer" l'exécution du code CATHARE

Solution ?



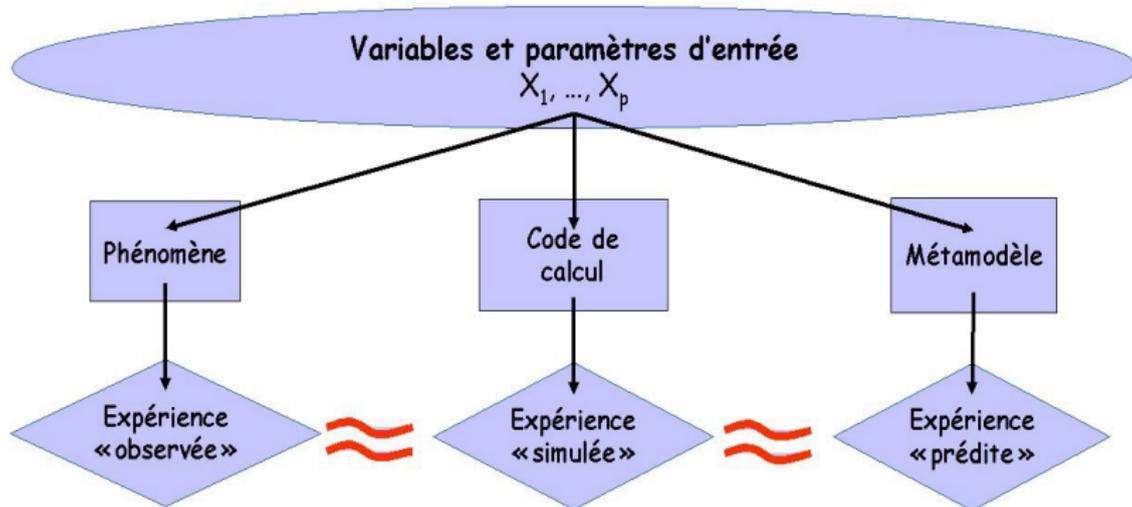
analyses statistiques sur le phénomène

Solution ?



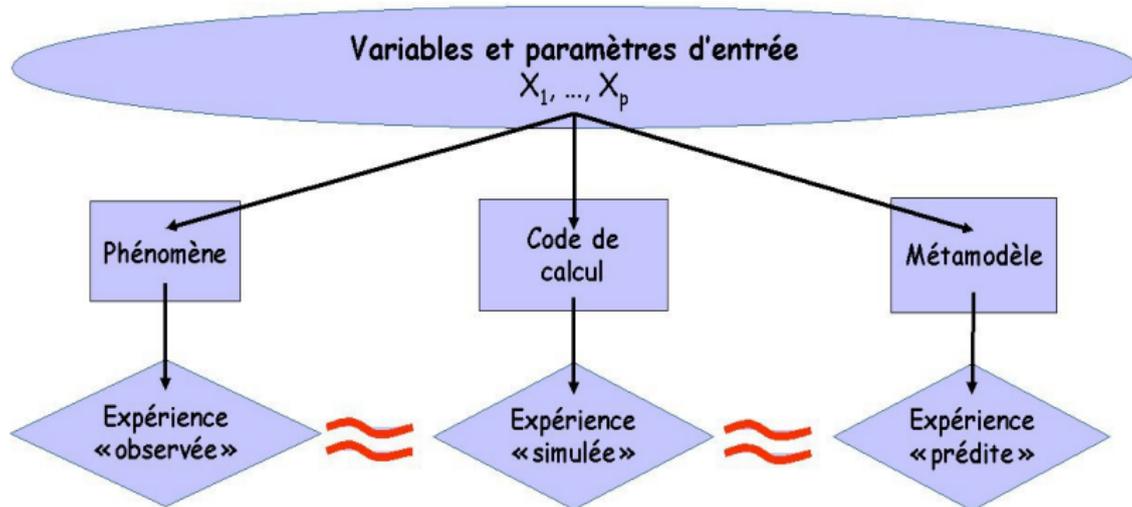
nécessitent analyses statistiques sur le phénomène
beaucoup de résultats de code

Solution : métamodèle



analyses statistiques sur le phénomène
nécessitent beaucoup de résultats de code
obtenus avec un métamodèle (= modèle du code).

Solution : métamodèle



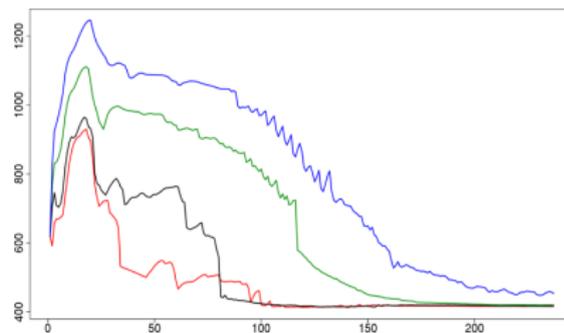
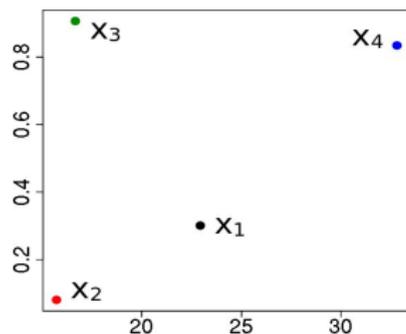
analyses statistiques sur le phénomène
nécessitent beaucoup de résultats de code
obtenus avec un métamodèle (= modèle du code).

Construire un modèle du code CATHARE

Reformulation finale

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.



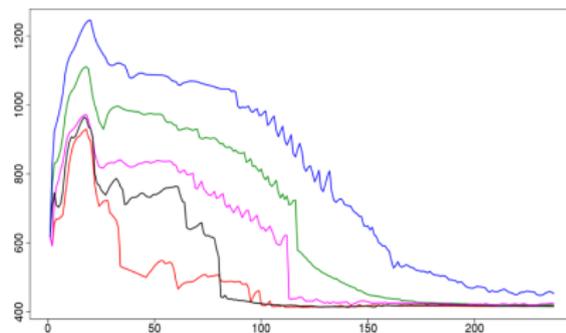
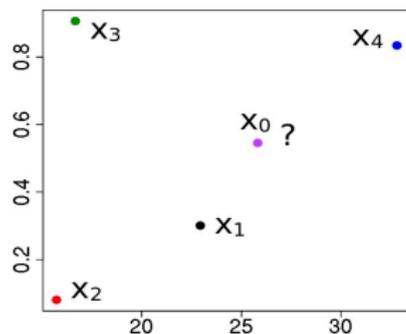
Reformulation finale

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.

Objectif = **prédiction** de données fonctionnelles :

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



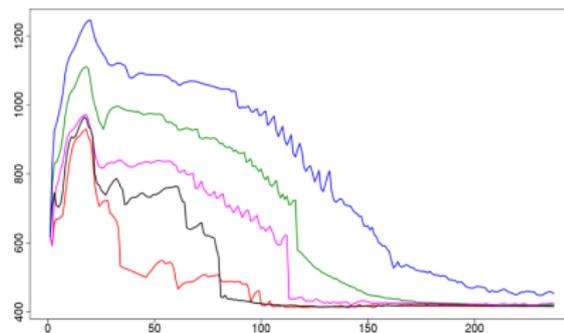
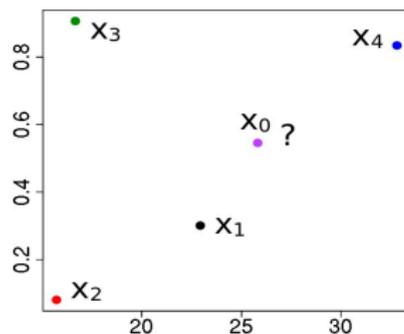
Reformulation finale

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.

Objectif = **prédiction** de données fonctionnelles :

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



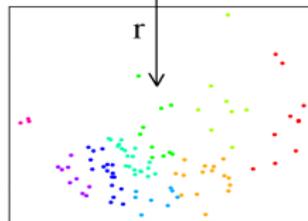
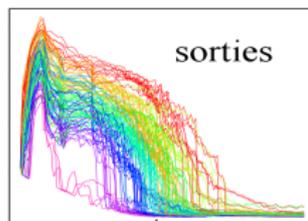
Apprentissage statistique "régression" $\mathbb{R}^p \rightarrow \mathcal{C}([a, b], \mathbb{R})$

Comment se ramener au cas "simple" $y_i \in \mathbb{R}^d$?

Comment se ramener au cas "simple" $y_i \in \mathbb{R}^d$?

- 1 réduction de la dimension :

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



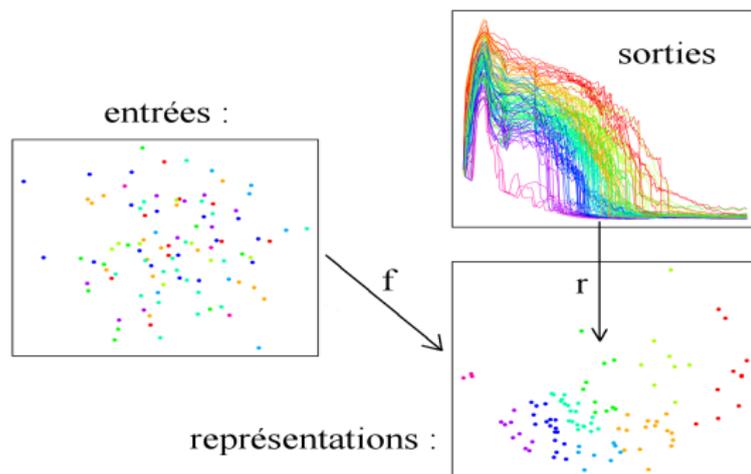
Comment se ramener au cas "simple" $y_i \in \mathbb{R}^d$?

- 1 réduction de la dimension :

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

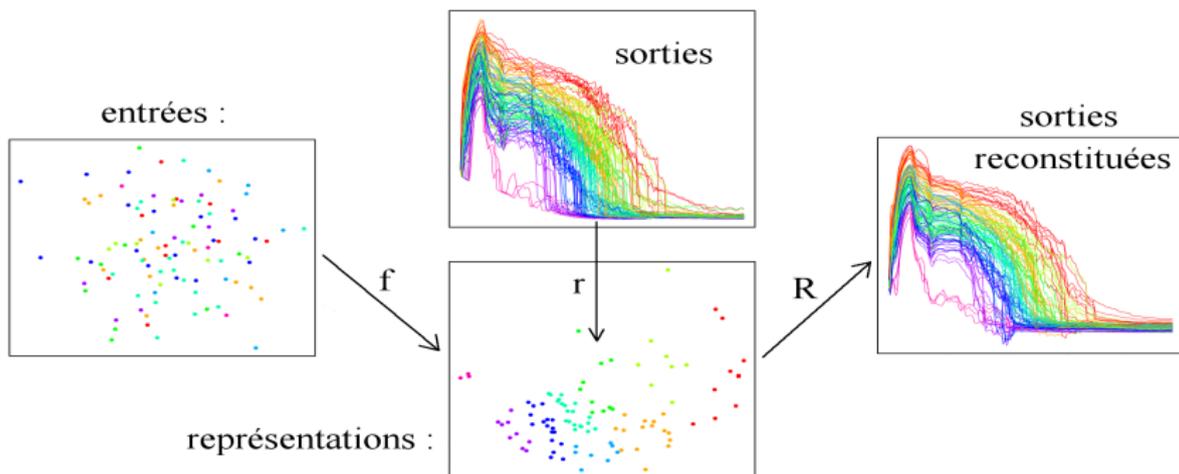
- 2 apprentissage statistique classique :

$$f : \mathbb{R}^p \text{ (entrées)} \rightarrow \mathbb{R}^d \text{ (sorties réduites) ;}$$



Comment se ramener au cas "simple" $y_i \in \mathbb{R}^d$?

- 1 réduction de la dimension :
 $r : \mathcal{C}([a, b], \mathbb{R}) \rightarrow \mathbb{R}^d$ (représentation) ;
- 2 apprentissage statistique classique :
 $f : \mathbb{R}^p$ (entrées) $\rightarrow \mathbb{R}^d$ (sorties réduites) ;
- 3 paramétrage de l'espace des sorties :
 $R : \mathbb{R}^d \rightarrow \mathcal{C}([a, b], \mathbb{R})$ (reconstruction).



Spécialisation du (méta)modèle

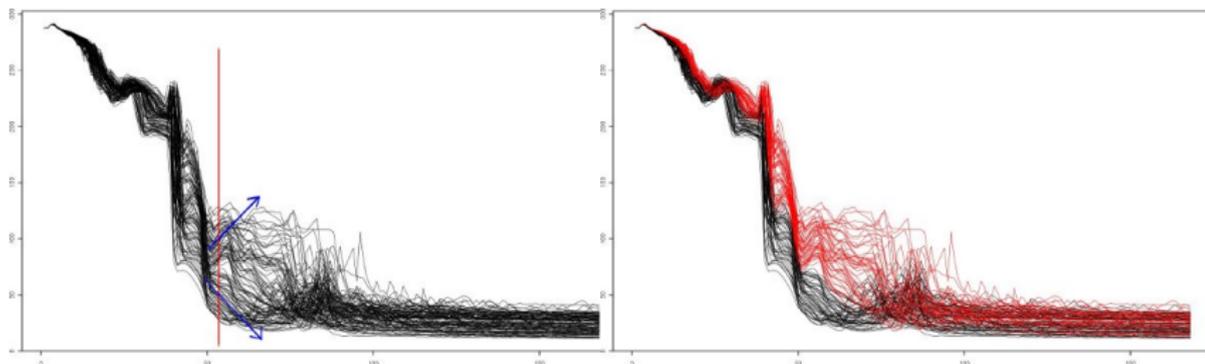


Figure: 100 transitoires de température

→ Différents types de comportements physiques.

Spécialisation du (méta)modèle

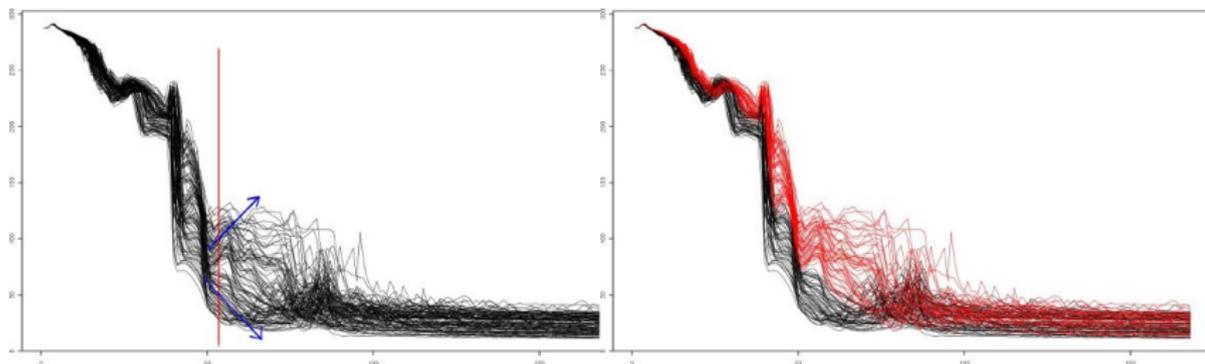


Figure: 100 transitoires de température

→ Différents types de comportements physiques.

Regroupement des courbes aux caractéristiques similaires

⇒ meilleure modélisation dans chaque cluster.

Spécialisation du (méta)modèle

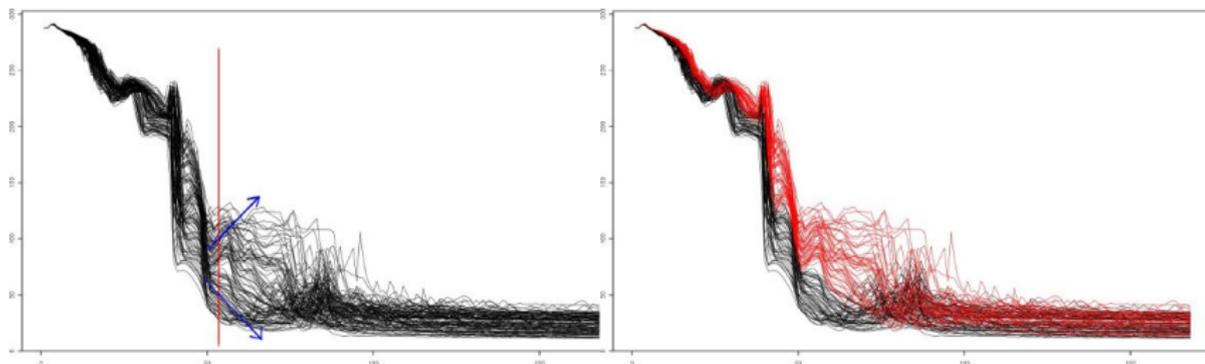


Figure: 100 transitoires de température

→ Différents types de comportements physiques.

Regroupement des courbes aux caractéristiques similaires

⇒ meilleure modélisation dans chaque cluster.

Classification des sorties *puis* des entrées.

Chronologie

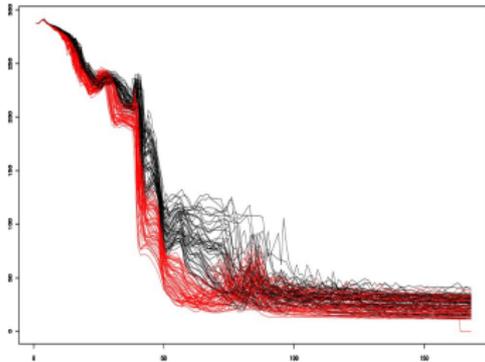


Figure: 100 transitoires de température en sortie (cathare)

- 1 Classification non supervisée des n données en K clusters C_j

Chronologie

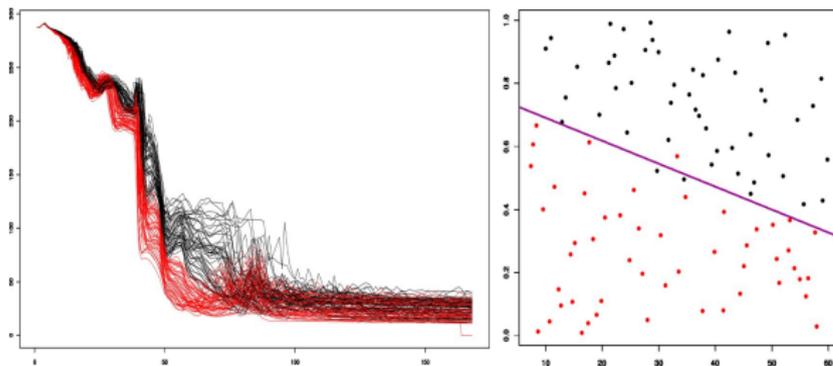


Figure: g. à d. : sorties fonctionnelles, scatterplot entrées 1 - 4

- 1 Classification non supervisée des n données en K clusters C_j
+ classification supervisée des entrées x_j .

Chronologie

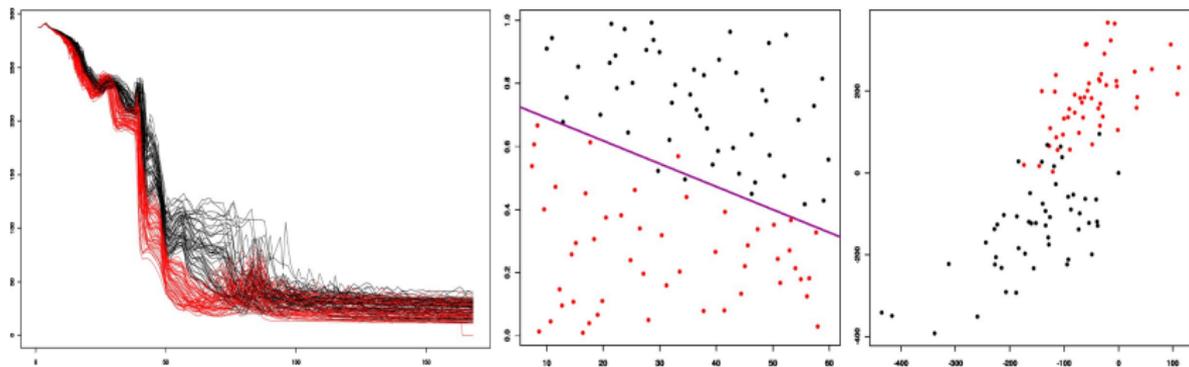


Figure: g. à d. : sorties fonc., entrées 1 - 4, représentation 2D des sorties

- 1 Classification non supervisée des n données en K clusters C_j
+ classification supervisée des entrées x_i .
- 2 Pour chaque cluster C_j ,
 - 1 réduction de la dimension : $r(y_i) = z_i$ représente y_i dans \mathbb{R}^d ;

Chronologie

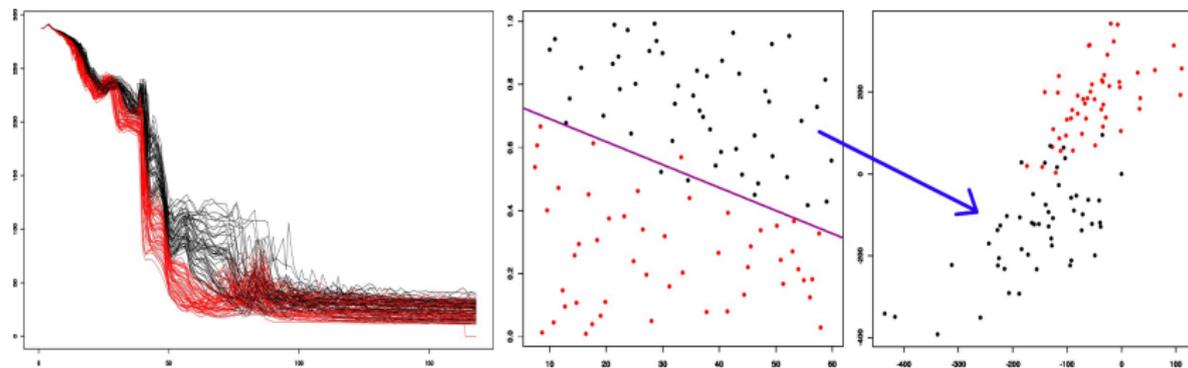


Figure: g. à d. : sorties fonc., entrées 1 - 4, rep. 2D des sorties

- 1 Classification non supervisée des n données en K clusters C_j
+ classification supervisée des entrées x_i .
- 2 Pour chaque cluster C_j ,
 - 1 réduction de la dimension : $r(y_i) = z_i$ représente y_i dans \mathbb{R}^d ;
 - 2 apprentissage d'une fonction de régression : $f(x_i) \simeq z_i$;

Chronologie

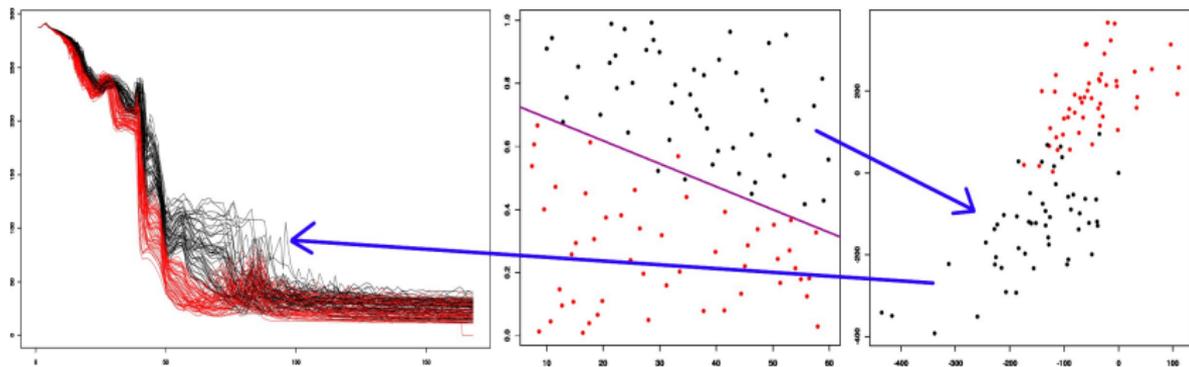


Figure: g. à d. : sorties fonc., entrées 1 - 4, rep. 2D des sorties

- 1 Classification non supervisée des n données en K clusters C_j
+ classification supervisée des entrées x_i .
- 2 Pour chaque cluster C_j ,
 - 1 réduction de la dimension : $r(y_i) = z_i$ représente y_i dans \mathbb{R}^d ;
 - 2 apprentissage d'une fonction de régression : $f(x_i) \simeq z_i$;
 - 3 apprentissage d'une fonction de reconstruction : $R(z_i) \simeq y_i$.

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d$, $x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d$, $x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Algorithme

Initialisation $r_i \leftarrow z_i^{(k)}$, $j \leftarrow 1$ (r_i = résidu d'approximation).

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d, x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Algorithme

Initialisation $r_i \leftarrow z_i^{(k)}, j \leftarrow 1$ (r_i = résidu d'approximation).

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

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d$, $x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;

J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Algorithme

Initialisation $r_i \leftarrow z_i^{(k)}$, $j \leftarrow 1$ (r_i = résidu d'approximation).

- 1 **Projection** : déterminer $\alpha_j \in \mathbb{R}^p$ direction de projection.
- 2 **Pursuit regression** : lissage non paramétrique sur les points $\langle \alpha_j, x_i \rangle, r_i$: $m_j(\langle \alpha_j, x_i \rangle) \simeq r_i$.

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d$, $x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;
J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Algorithme

Initialisation $r_i \leftarrow z_i^{(k)}$, $j \leftarrow 1$ (r_i = résidu d'approximation).

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

Régression $\mathbb{R}^p \rightarrow \mathbb{R}^d$, $x \mapsto (f_1(x), \dots, f_d(x))$

Modèle utilisé : *Projection Pursuit Regression* ;
J. H. Friedman & W. Stuetzle, (1981).

Données : $(x_1, z_1^{(k)} = f_k(x_1)), \dots, (x_n, z_n^{(k)} = f_k(x_n))$. $\hat{f}_k(x) = ?$

Algorithme

Initialisation $r_i \leftarrow z_i^{(k)}$, $j \leftarrow 1$ (r_i = résidu d'approximation).

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

$$\text{Prédiction : } \hat{z}^{(k)} = \sum_{j=1}^q m_j(\langle \alpha_j, x \rangle)$$

- 1 Clustering des entrées-sorties
- 2 Réduction de la dimension
- 3 Estimation de la dimension
- 4 Applications

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes

- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning

- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes

- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Première mesure de comparaison

Dans \mathbb{R}^p : distance euclidienne.

Première mesure de comparaison

Dans \mathbb{R}^p : distance euclidienne.

Dans $\mathcal{C}([a, b])$, plusieurs possibilités :

- distances L^q ;
- (semi-)distances basées sur les dérivées ;
- (dis)similarités basées sur les formes ;

Première mesure de comparaison

Dans \mathbb{R}^p : distance euclidienne.

Dans $\mathcal{C}([a, b])$, plusieurs possibilités :

- distances L^q ;
- (semi-)distances basées sur les dérivées ;
- (dis)similarités basées sur les formes ;

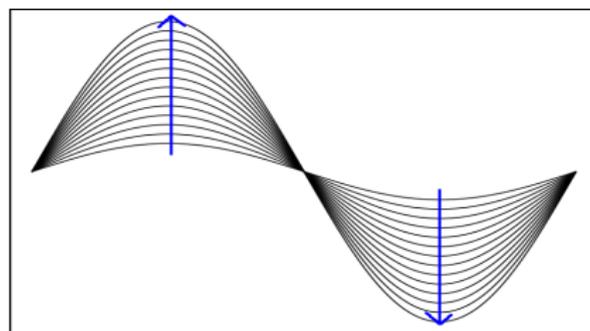


Figure: Courbes similaires reliées par un chemin

⇒ Distance L^2 en général suffisante.

Représentation des sorties dans un graphe

Graphe = (V, E) où

- V = ensemble des sorties y_i = "sommets" ;
- E = relations inter-sommets = "arêtes".
 $(y_i, y_j) \in E$ ssi. y_j proche de y_i .

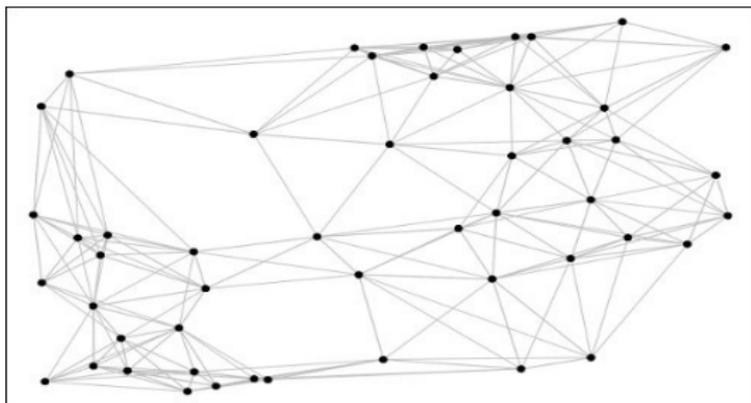


Figure: Représentation des sorties dans un graphe ;
sommets = y_i , arêtes = distances.

Représentation des sorties dans un graphe

Graphe = (V, E) où

- V = ensemble des sorties y_i = "sommets" ;
- E = relations inter-sommets = "arêtes".
 $(y_i, y_j) \in E$ ssi. y_j proche de y_i .

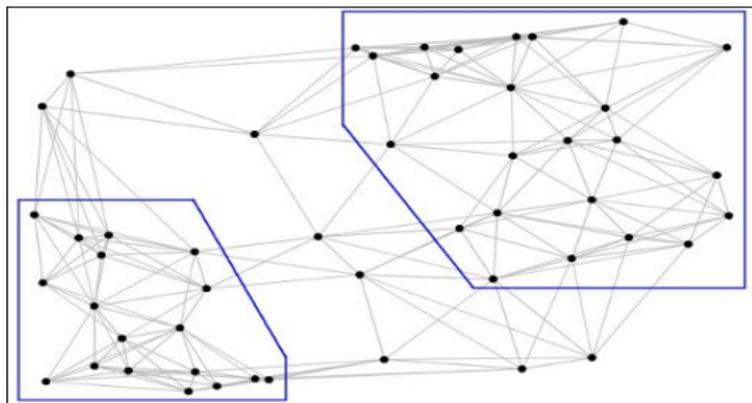


Figure: En bleu : deux clusters (visuels) ;

Représentation des sorties dans un graphe

Graphe = (V, E) où

- V = ensemble des sorties y_i = "sommets" ;
- E = relations inter-sommets = "arêtes".
 $(y_i, y_j) \in E$ ssi. y_j proche de y_i .

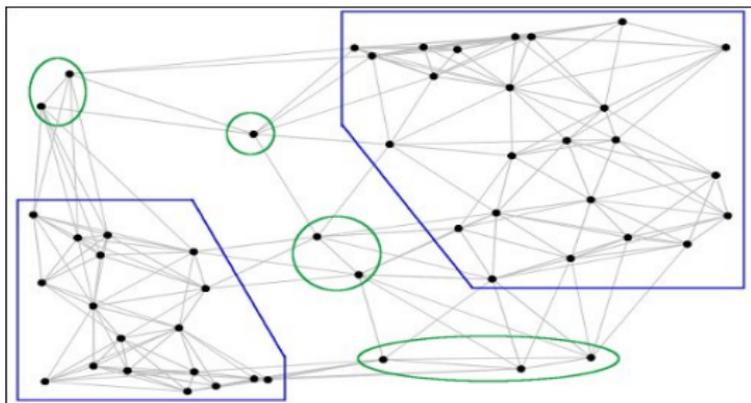


Figure: En vert : points "isolés" ;

Marche aléatoire

"A et B sont très proches"

\simeq "Il y a beaucoup de courts chemins de A à B"

\simeq "Une marche aléatoire issue de A, passant par B, reviendra vite vers A".

Marche aléatoire

"A et B sont très proches"

\simeq "Il y a beaucoup de courts chemins de A à B"

\simeq "Une marche aléatoire issue de A, passant par B, reviendra vite vers A".

Marche aléatoire dans le graphe de voisinage

Transition de y_i vers y_j avec probabilité

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

σ_i : paramètre local dépendant de k et des distances aux voisins.

Distance "temps d'aller-retour"

Principal résultat (M. Saerens et al. 2004)

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.

Distance "temps d'aller-retour"

Principal résultat (M. Saerens et al. 2004)

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 :

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

Distance "temps d'aller-retour"

Principal résultat (M. Saerens et al. 2004)

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 :

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

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

avec L^+ pseudo-inverse de L .

- 1 Clustering des entrées-sorties
 - Distance
 - **Classification ascendante hiérarchique (CAH)**
 - Détermination du nombre de classes

- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning

- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes

- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Fonctionnement

Algorithme déjà conçu pour une matrice de distances.

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

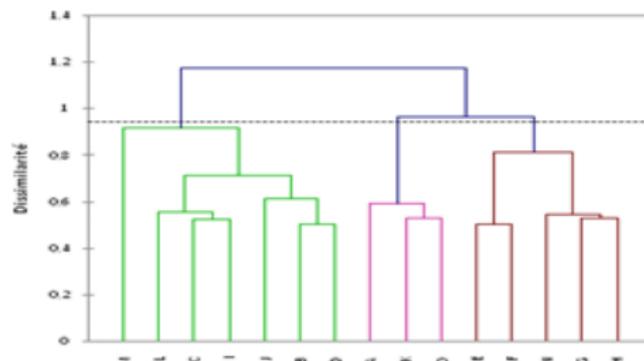


Figure: Dendrogramme résultat d'une CAH.

Fonctionnement

Algorithme déjà conçu pour une matrice de distances.

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

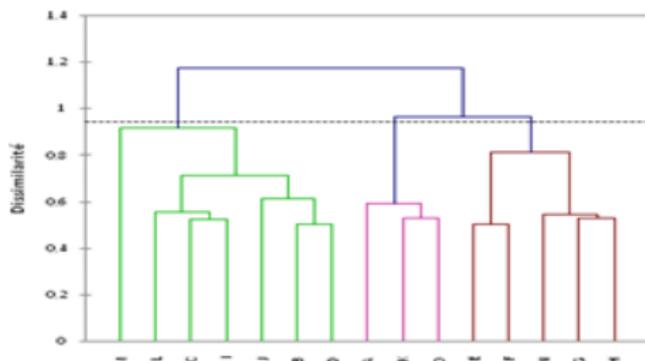


Figure: Dendrogramme résultat d'une CAH.

Résultat : suite de partitions emboîtées

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

Comparaisons inter-clusters

- 1 Définition d'une distance d dans $\{y_1, \dots, y_n\}$;
- 2 Définition d'une distance entre les groupes à partir de d .

Comparaisons inter-clusters

- 1 Définition d'une distance d dans $\{y_1, \dots, y_n\}$;
- 2 Définition d'une distance entre les groupes à partir de d .

1 : distance ECT calculée dans le graphe.

2 : distance de Ward, définie par

$$d_W(A, B) = \sqrt{\frac{|A||B|}{|A| + |B|}} d(g_A, g_B)$$

→ minimise la variation d'inertie

⇒ favorise les partitions "stables"

Exemple

240 courbes dans 3 clusters :

$$f_{\alpha,\beta,\gamma}(t) = \left(\frac{\sin \alpha t}{t} + e^{-\beta t} \right) \cos \gamma t$$

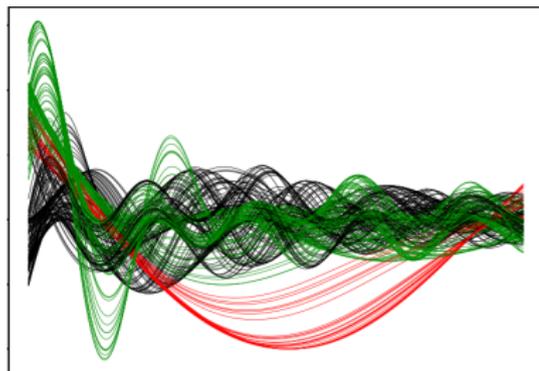


Figure: CAH + dist. L^2

Précision : \simeq 50%

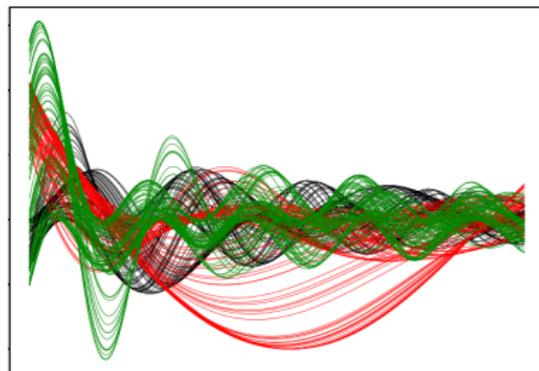


Figure: CAH + DECT

Précision : 100%

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Stabilité

Une légère modification du jeu de données ne doit pas changer sensiblement le résultat.

Stabilité

Une légère modification du jeu de données ne doit pas changer sensiblement le résultat.

Procédure I (P_1) = répéter N fois ...

- 1 sous-échantillonner U et U' de tailles αn ;

Stabilité

Une légère modification du jeu de données ne doit pas changer sensiblement le résultat.

Procédure I (P_1) = répéter N fois ...

- 1 sous-échantillonner U et U' de tailles αn ;
- 2 appliquer AC à U et $U' \Rightarrow$ partitions P et P' ;
[AC = algorithme de clustering (CAH + DECT)].

Stabilité

Une légère modification du jeu de données ne doit pas changer sensiblement le résultat.

Procédure I (P_1) = répéter N fois ...

- 1 sous-échantillonner U et U' de tailles αn ;
- 2 appliquer AC à U et $U' \Rightarrow$ partitions P et P' ;
[AC = algorithme de clustering (CAH + DECT)].
- 3 $\tau \leftarrow \tau + \text{similarité}(P, P')$.

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Procédure II (P_2) = répéter N fois ...

- 1 diviser les données en E (entraînement) et T (test);

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Procédure II (P_2) = répéter N fois ...

- 1 diviser les données en E (entraînement) et T (test);
- 2 appliquer AC à $T \Rightarrow$ partition P ;

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Procédure II (P_2) = répéter N fois ...

- 1 diviser les données en E (entraînement) et T (test);
- 2 appliquer AC à $T \Rightarrow$ partition P ;
- 3 appliquer AC à E (entraînement) \Rightarrow partition H ;

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Procédure II (P_2) = répéter N fois ...

- 1 diviser les données en E (entraînement) et T (test);
- 2 appliquer AC à $T \Rightarrow$ partition P ;
- 3 appliquer AC à E (entraînement) \Rightarrow partition H ;
- 4 classification supervisée $E \rightarrow H \Rightarrow$ partition P' par prédiction des étiquettes de T (test);

Capacité de prédiction

Les classes des entrées doivent être prédites avec précision.

Procédure II (P_2) = répéter N fois ...

- 1 diviser les données en E (entraînement) et T (test);
- 2 appliquer AC à $T \Rightarrow$ partition P ;
- 3 appliquer AC à E (entraînement) \Rightarrow partition H ;
- 4 classification supervisée $E \rightarrow H \Rightarrow$ partition P' par prédiction des étiquettes de T (test);
- 5 $\tau \leftarrow \tau + \text{similarité}(P, P')$.

Bilan

Plus il y a de groupes, plus le modèle peut être précis. Donc :

Algorithme = incrémenter K jusqu'à voir $\tau < \tau_S$ ou $\tau < \tau_C$.

Avec K = nombre de clusters.

Bilan

Plus il y a de groupes, plus le modèle peut être précis. Donc :

Algorithme = incrémenter K jusqu'à voir $\tau < \tau_S$ ou $\tau < \tau_C$.

Avec K = nombre de clusters.

Puis : même méthodologie dans chaque groupe en entrée.

...éventuellement en changeant AC .

Bilan

Résumé

- 1 Graphe représentant les similarités locales.

Bilan

Résumé

- 1 Graphe représentant les similarités locales.
- 2 Marche aléatoire → matrice des distances.

Bilan

Résumé

- 1 Graphe représentant les similarités locales.
- 2 Marche aléatoire → matrice des distances.
- 3 Clustering hiérarchique → suite de partitions emboîtées.

Bilan

Résumé

- 1 Graphe représentant les similarités locales.
- 2 Marche aléatoire → matrice des distances.
- 3 Clustering hiérarchique → suite de partitions emboîtées.
- 4 Tests P_1 et P_2 jusqu'à K maximum.

Bilan

Résumé

- 1 Graphe représentant les similarités locales.
- 2 Marche aléatoire → matrice des distances.
- 3 Clustering hiérarchique → suite de partitions emboîtées.
- 4 Tests P_1 et P_2 jusqu'à K maximum.

Contributions

- Réglage local du paramètre d'échelle.
- Algorithme de détermination du nombre de clusters.

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - **Approches linéaires**
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

ACP fonctionnelle

Représentation $r(y) = (\langle y, e_1 \rangle, \dots, \langle y, e_d \rangle) = z$.

Reconstruction $R(z) = \sum_{j=1}^d z_j e_j \simeq y$.

ACP fonctionnelle

Représentation $r(y) = (\langle y, e_1 \rangle, \dots, \langle y, e_d \rangle) = z$.

Reconstruction $R(z) = \sum_{j=1}^d z_j e_j \simeq y$.

Courbes discrétisées en lignes : $A = \begin{pmatrix} y_{11} & \dots & y_{1D} \\ \vdots & \vdots & \vdots \\ y_{n1} & \dots & y_{nD} \end{pmatrix}$

Représentation L^2 -optimale

Décomposition en valeurs singulières : $A = {}^t U \Sigma V$.

ACP fonctionnelle

Représentation $r(y) = (\langle y, e_1 \rangle, \dots, \langle y, e_d \rangle) = z$.

Reconstruction $R(z) = \sum_{j=1}^d z_j e_j \simeq y$.

Courbes discrétisées en lignes : $A = \begin{pmatrix} y_{11} & \dots & y_{1D} \\ \vdots & \vdots & \vdots \\ y_{n1} & \dots & y_{nD} \end{pmatrix}$

Représentation L^2 -optimale

Décomposition en valeurs singulières : $A = {}^t U \Sigma V$.

$\{e_1, \dots, e_d\}$ = premières colonnes de V .

Autres bases

Polynômes, splines; Fourier, ondelettes, ...

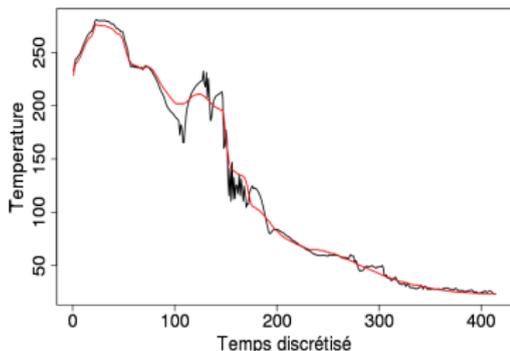


Figure: Approx. ACP

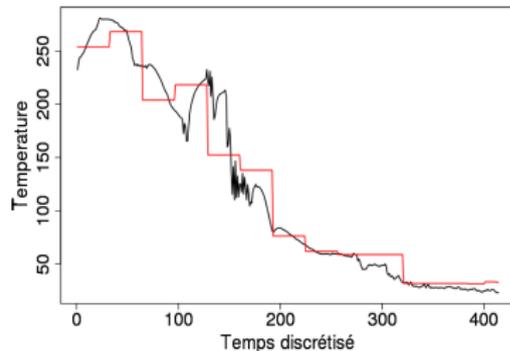


Figure: Approx. Haar (ondelettes)

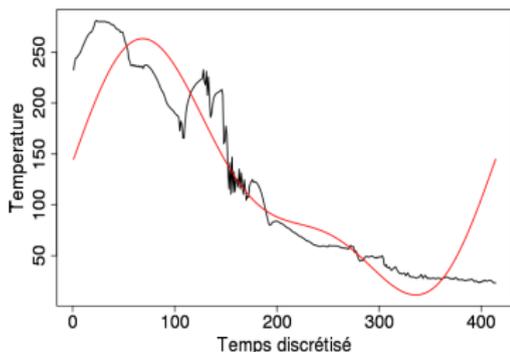


Figure: Approx. Fourier

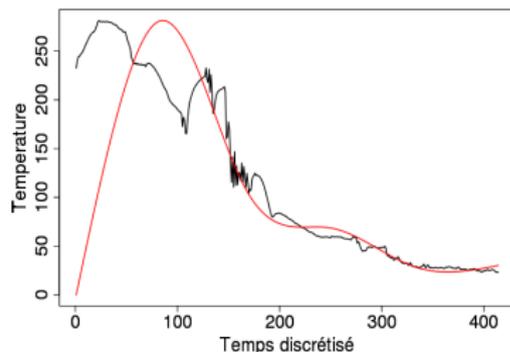


Figure: Approx. B-splines

Structure des sorties = ?

Structure des sorties = ?

Variété (connexe)

Espace topologique connexe séparé, à base dénombrable d'ouverts
localement homéomorphe à \mathbb{R}^d .

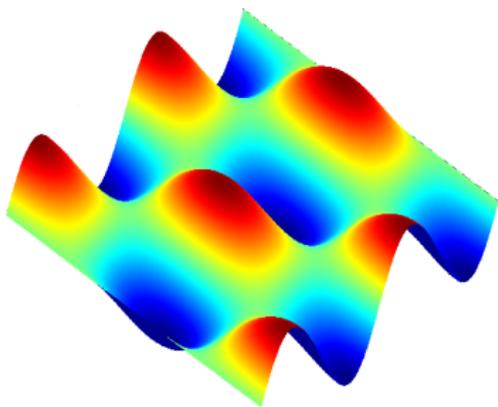


Figure: Visualisation d'une variété

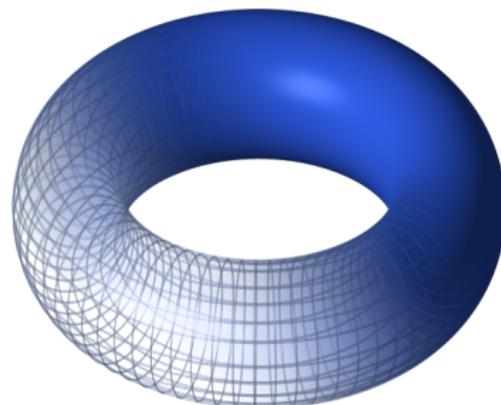


Figure: Tore 2D dans \mathbb{R}^3

Structure des sorties = ?

Variété (connexe)

Espace topologique connexe séparé, à base dénombrable d'ouverts
localement homéomorphe à \mathbb{R}^d .

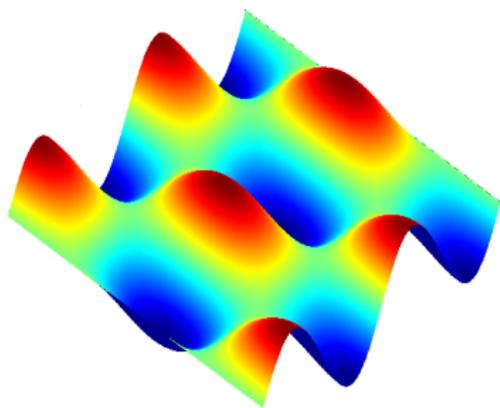


Figure: Visualisation d'une variété

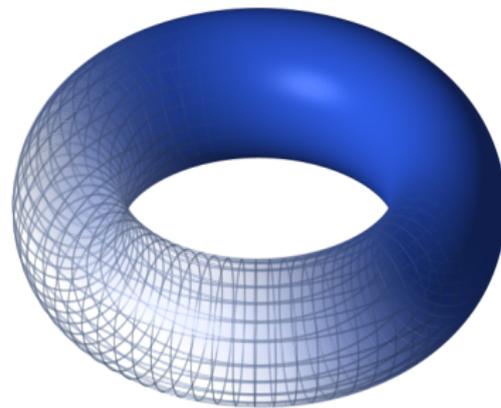


Figure: Tore 2D dans \mathbb{R}^3

Hypothèse : sorties du code sur une variété (lisse ...).

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes

- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning

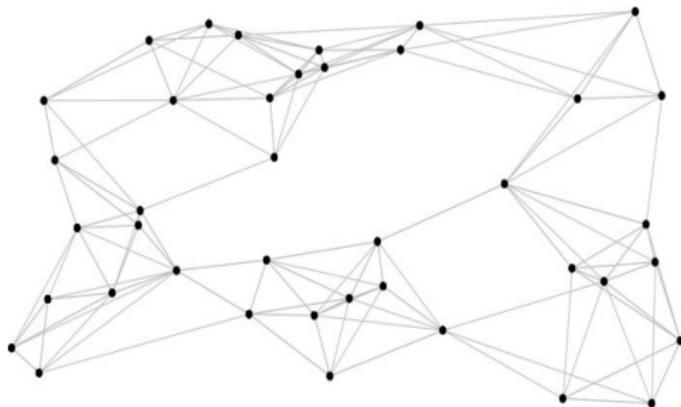
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes

- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Étape préliminaire (algorithme de Lin et al., 2006)

Représentation des sorties
dans un graphe ;

sommets = courbes,
arêtes = distances.



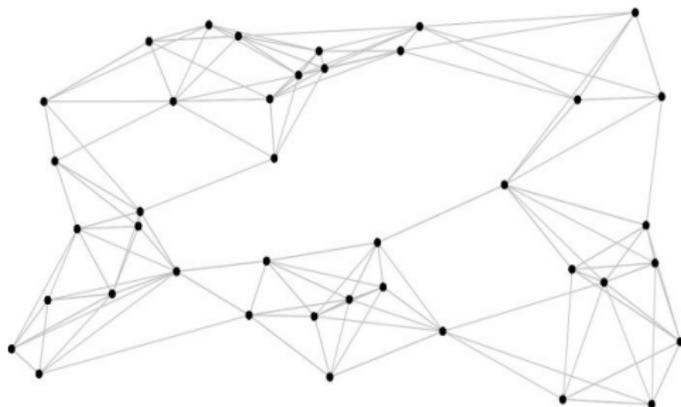
y_j se connecte à y_i si :

- $\|y_i - y_j\| \leq \epsilon \rightarrow \epsilon$ -graphe (symétrique) ;
- y_j est parmi les k plus proches voisins de y_i (norme L^2 p.ex.).

Étape préliminaire (algorithme de Lin et al., 2006)

Représentation des sorties
dans un graphe ;

sommets = courbes,
arêtes = distances.

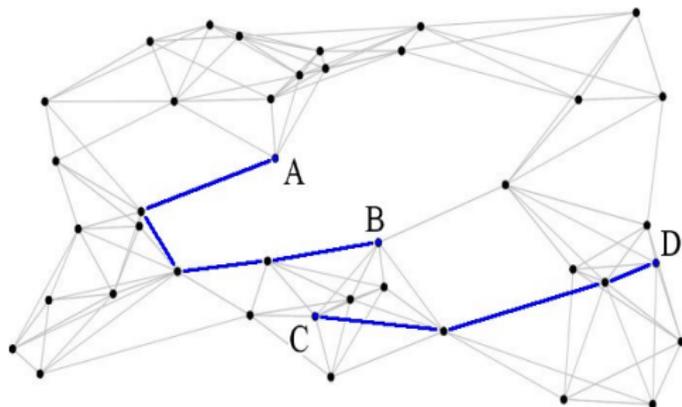


y_j se connecte à y_i si :

- $\|y_i - y_j\| \leq \epsilon \rightarrow \epsilon$ -graphe (symétrique) ;
- y_j est parmi les k plus proches voisins de y_i (norme L^2 p.ex.).

Étape préliminaire (algorithme de Lin et al., 2006)

En bleu :
deux plus courts chemins ;



y_j se connecte à y_i si :

- $\|y_i - y_j\| \leq \epsilon \rightarrow \epsilon$ -graphe (symétrique) ;
- y_j est parmi les k plus proches voisins de y_i (norme L^2 p.ex.).

Recherche de y_0 (origine)

Pour chaque sommet (= courbe y_i), calculer la somme S des plus courts chemins à tous les autres sommets. Choisir $y_0 = \arg \min S$

Étape locale I

Base locale $Q_0 = (e_1, \dots, e_d)$ de l'espace tangent en y_0
(ACP sur un **voisinage de y_0**)

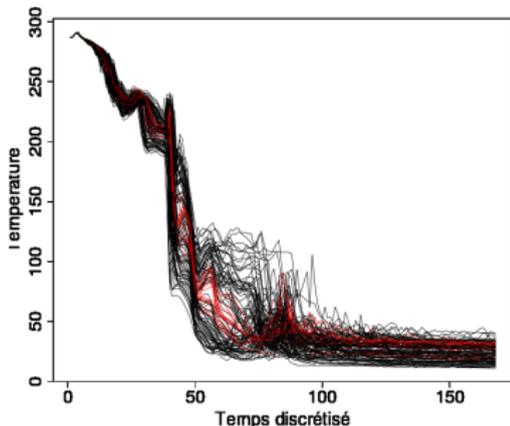


Figure: 15 voisins de y_0

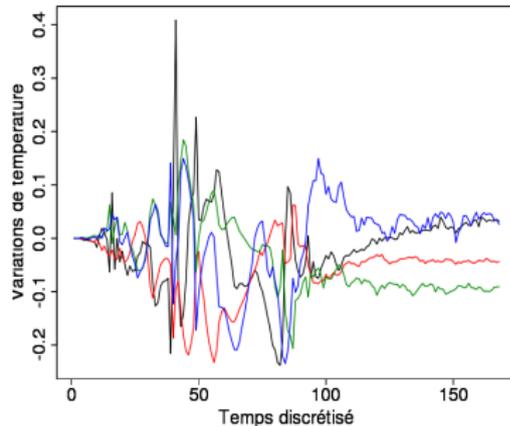


Figure: Base locale en y_0 ($d = 4$)

Étape locale II

Coordonnées réduites z_i calculées par projection sur Q_0 ,

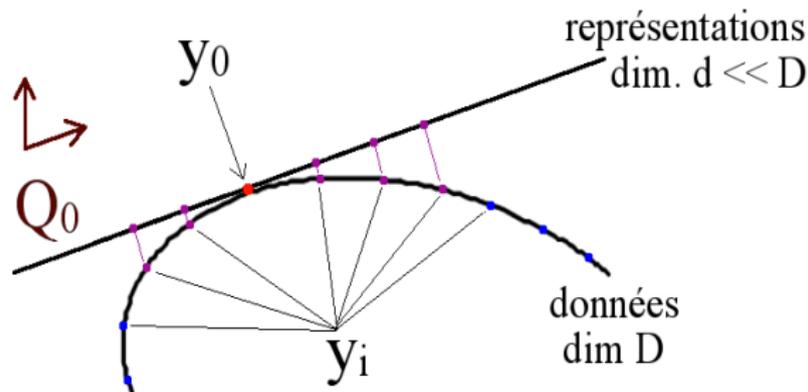


Figure: Coordonnées locales z_i sur l'espace tangent

Étape locale II

Coordonnées réduites z_i calculées par projection sur Q_0 ,
 + normalisation pour satisfaire la contrainte $\|y_i - y_0\| = \|z_i - z_0\|$.

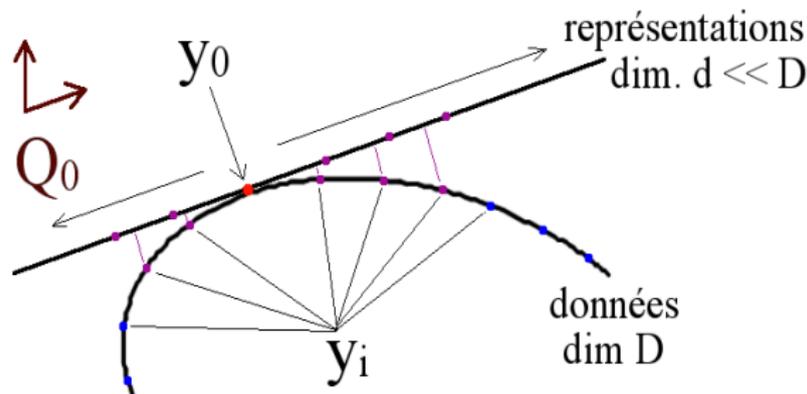


Figure: Normalisation des coordonnées z_i

y_i "loin" de y_0

- y_p = prédécesseur de y_i sur un plus court chemin depuis y_0
- y_{i1}, \dots, y_{im} = voisins de y_p dont les coords. z_{ij} sont connues

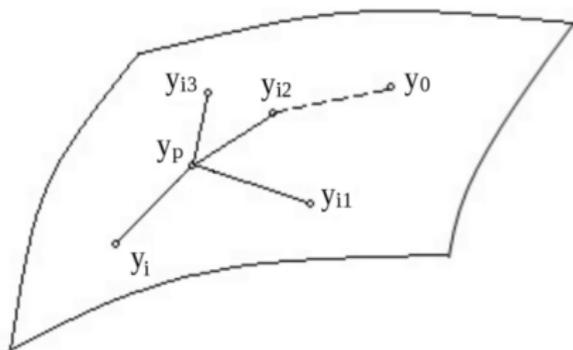


Figure: Données y_i en dim. D

y_i "loin" de y_0

- y_p = prédécesseur de y_i sur un plus court chemin depuis y_0
- y_{i_1}, \dots, y_{i_m} = voisins de y_p dont les coords. z_{ij} sont connues

$z_i = r(y_i)$ calculé en..

- conservant au mieux les angles : $\cos \widehat{z_i z_p z_{ij}} \simeq \cos \widehat{y_i y_p y_{ij}}$
avec $\cos \widehat{u_1 u_2 u_3} = \frac{\langle u_1 - u_2, u_3 - u_2 \rangle}{\|u_1 - u_2\| \|u_3 - u_2\|}$;
- sous la contrainte de normalisation $\|y_i - y_p\| = \|z_i - z_p\|$.

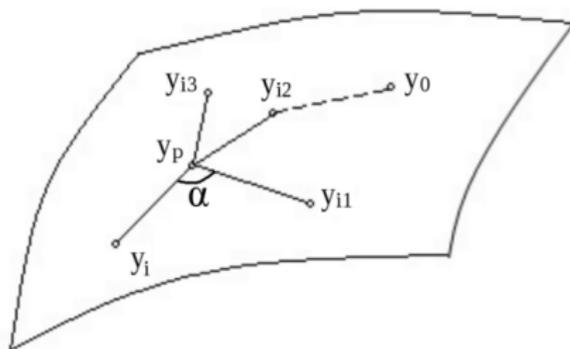


Figure: Données y_i en dim. D

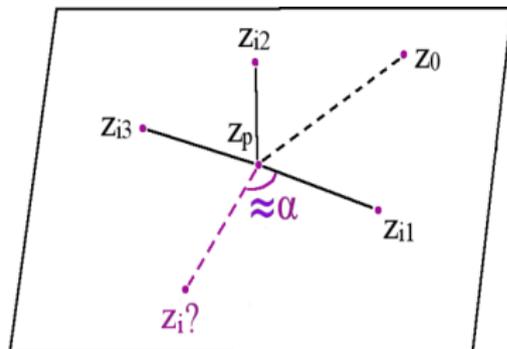


Figure: $z_i = r(y_i)$ en dim. $d \ll D$

Exemple

Swissroll fonctionnel :

$$f_{\alpha,\beta,\gamma}(x \in [0, 4\pi]) =$$

$$\alpha \cos x + \beta \sin x + \frac{\gamma}{2} \sin 2x$$

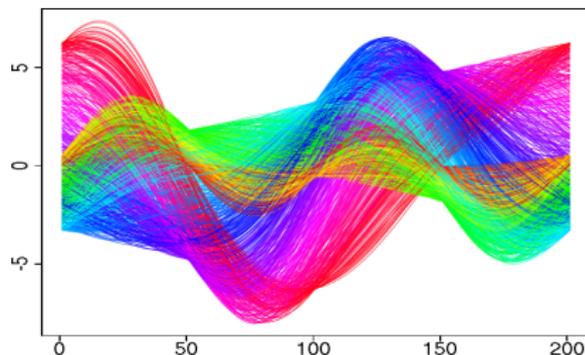


Figure: Swissroll fonctionnel

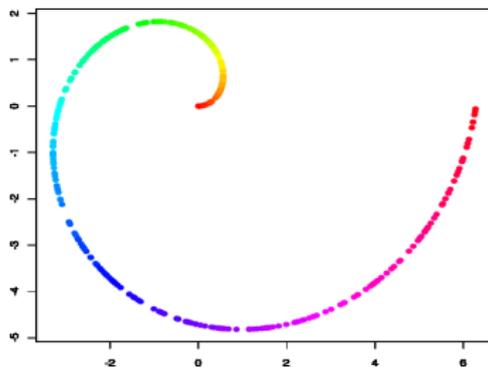


Figure: représentation ACP(F)

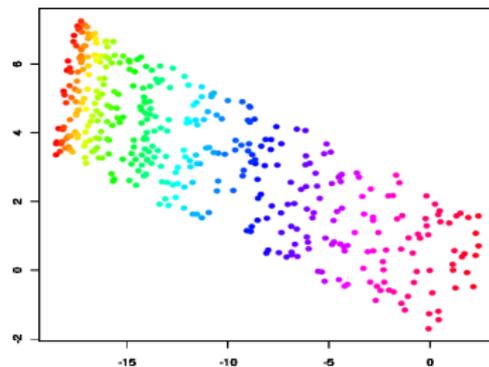


Figure: représentation RML

Bilan

Résumé

Réduction de dimension pour permettre la régression vectorielle.

Bilan

Résumé

Réduction de dimension pour permettre la régression vectorielle.

- 1 Construction du graphe de visibilité des sorties y_j .

Bilan

Résumé

Réduction de dimension pour permettre la régression vectorielle.

- 1 Construction du graphe de visibilité des sorties y_j .
- 2 Projection sur l'espace tangent \rightarrow coords. locales en y_0 .

Bilan

Résumé

Réduction de dimension pour permettre la régression vectorielle.

- 1 Construction du graphe de visibilité des sorties y_j .
- 2 Projection sur l'espace tangent \rightarrow coords. locales en y_0 .
- 3 Propagation itérative de contraintes \rightarrow autres coords.

Bilan

Résumé

Réduction de dimension pour permettre la régression vectorielle.

- 1 Construction du graphe de visibilité des sorties y_j .
- 2 Projection sur l'espace tangent \rightarrow coords. locales en y_0 .
- 3 Propagation itérative de contraintes \rightarrow autres coords.

Contributions

- Implémentation d'une méthode existante dans un contexte nouveau.
- Comparaison à d'autres techniques de réduction de dimension.

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Première approche

Courbes (discrétisées) : $y_1, \dots, y_n \in V \subset \mathcal{C}([a, b])$.

Première approche

Courbes (discrétisées) : $y_1, \dots, y_n \in V \subset \mathcal{C}([a, b])$.

$H_1 : V =$ espace vectoriel

- 1 recherche d'une famille génératrice \mathcal{G} dans V ;
- 2 orthonormalisation de \mathcal{G} (via SVD p.ex.) en \mathcal{G}' .

\mathcal{G}' bon. de V , de dimension $d = |\mathcal{G}'|$.

Première approche

Courbes (discrétisées) : $y_1, \dots, y_n \in V \subset \mathcal{C}([a, b])$.

$H_1 : V = \text{espace vectoriel}$

- 1 recherche d'une famille génératrice \mathcal{G} dans V ;
- 2 orthonormalisation de \mathcal{G} (via SVD p.ex.) en \mathcal{G}' .

\mathcal{G}' bon. de V , de dimension $d = |\mathcal{G}'|$.

$H_2 : V = \text{variété}$

L'espace tangent en $y \in V$ est isomorphe à \mathbb{R}^d .

Approximation : $\{y + k \text{ plus proches voisins}\} \simeq \text{sev.}$

\Rightarrow méthode précédente localement, puis moyenne.

Difficulté

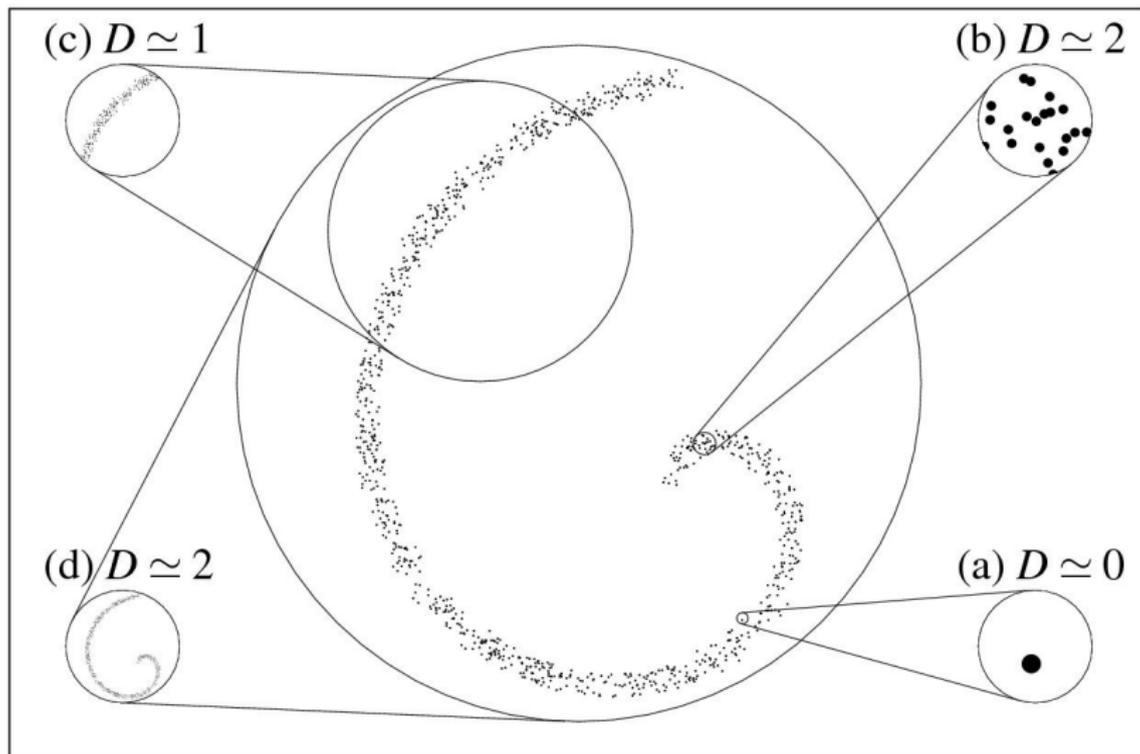


Figure: Spirale bruitée à différentes échelles

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Algorithme de Levina et Bickel (2005) - hypothèses

Hypothèse 1

Le nombre de points dans $S(y, \rho) \cap \{y_i\}_{i=1, \dots, n}$ correspond à un processus de Poisson inhomogène indexé par ρ .

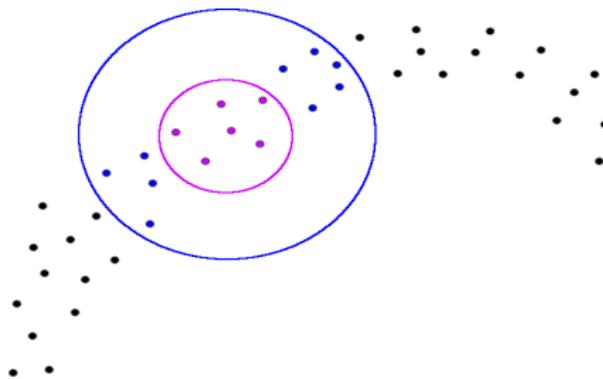


Figure: Illustration sur une variété 2D

Algorithme de Levina et Bickel (2005) - hypothèses

Hypothèse 1

Le nombre de points dans $S(y, \rho) \cap \{y_i\}_{i=1, \dots, n}$ correspond à un processus de Poisson inhomogène indexé par ρ .

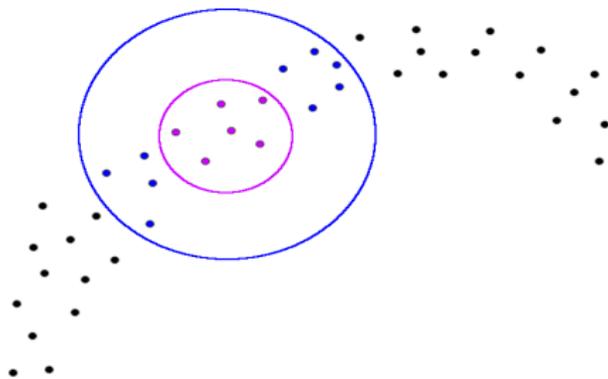


Figure: Illustration sur une variété 2D

Hypothèse 2

Approximation de la densité à l'ordre 0 : $f(y + \epsilon) \simeq f(y)$.

Algorithme de Levina et Bickel (2005) - calcul de d

Taux du processus :

$$\lambda(y, \rho) \simeq f(y) V_d d \rho^{d-1}$$

(densité \times surface de $S(y, \rho)$).

Algorithme de Levina et Bickel (2005) - calcul de d

Taux du processus :

$$\lambda(y, \rho) \simeq f(y) V_d d \rho^{d-1}$$

(densité \times surface de $S(y, \rho)$).

Maximisation de la log-vraisemblance :

$$d_y \simeq \frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{\rho_k(y)}{\rho_j(y)}$$

$$\hat{d} = \text{moyenne des } d_{y_i}$$

Estimation géométrique de d - introduction

$(m-)$ simplexe

Enveloppe convexe de $m + 1$ points formant un repère affine dans un ev. de dimension $\geq m$.

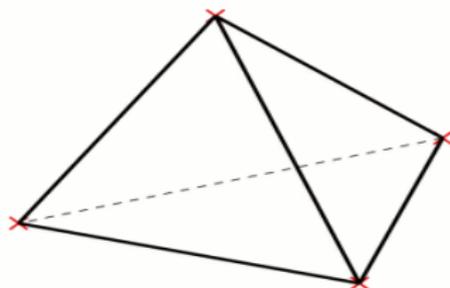


Figure: Tétrahédre = 3-simplexe dans \mathbb{R}^3

Estimation géométrique de d - introduction

(m) -simplexe

Enveloppe convexe de $m + 1$ points formant un repère affine dans un ev. de dimension $\geq m$.

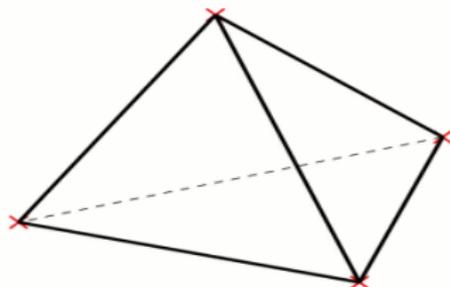


Figure: Tétrahédre = 3-simplexe dans \mathbb{R}^3

Observation : $\exists m$ -simplexes ssi. $m \leq d$.

\Rightarrow Idée : augmenter m jusqu'à ce que tous les m -simplexes soient de volume négligeable.

Estimation géométrique de d - introduction

(m) -simplexe

Enveloppe convexe de $m + 1$ points formant un repère affine dans un ev. de dimension $\geq m$.

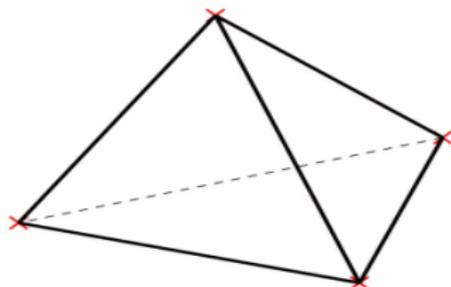


Figure: Tétrahédre = 3-simplexe dans \mathbb{R}^3

Observation : $\exists m$ -simplexes ssi. $m \leq d$.

\Rightarrow Idée : augmenter m jusqu'à ce que tous les m -simplexes soient de **volume négligeable** (?)

Estimation géométrique de d - algorithme

Estimer la densité de probabilité f des distances intra-voisines

Estimation géométrique de d - algorithme

Estimer la densité de probabilité f des distances intra-voisines

Pour d allant de 2 à d_{max} :

- 1 échantillonner N volumes de d -simplexes aléatoires dont les longueurs des côtés suivent f ;

Estimation géométrique de d - algorithme

Estimer la densité de probabilité f des distances intra-voisines

Pour d allant de 2 à d_{max} :

- 1 échantillonner N volumes de d -simplexes aléatoires dont les longueurs des côtés suivent f ;
- 2 estimer la densité de probabilité v des volumes des simplexes engendrés par $d + 1$ points aléatoires dans un voisinage ;

Estimation géométrique de d - algorithme

Estimer la densité de probabilité \mathbf{f} des distances intra-voisines

Pour d allant de 2 à d_{max} :

- 1 échantillonner N volumes de d -simplexes aléatoires dont les longueurs des côtés suivent \mathbf{f} ;
- 2 estimer la densité de probabilité \mathbf{v} des volumes des simplexes engendrés par $d + 1$ points aléatoires dans un voisinage ;
- 3 si \mathbf{f} et \mathbf{v} sont trop différentes, retourner $d - 1$.

Exemple : 10-twist Möbius

10 torsions appliquées à un ruban, puis recollage.

$$x_1(u, v) = \left(1 + \frac{u}{2} \cos\left(\frac{k}{2}v\right)\right) \cos v, \quad x_2(u, v) = \left(1 + \frac{u}{2} \cos\left(\frac{k}{2}v\right)\right) \sin v,$$

$$x_3(u, v) = \frac{u}{2} \sin\left(\frac{k}{2}v\right)$$

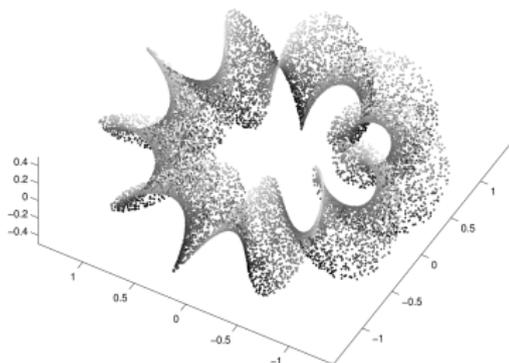


Figure: 16000 points sur le "10-Ruban" de Möbius

Exemple : 10-twist Möbius

10 torsions appliquées à un ruban, puis recollage.

$$x_1(u, v) = \left(1 + \frac{u}{2} \cos\left(\frac{k}{2}v\right)\right) \cos v, \quad x_2(u, v) = \left(1 + \frac{u}{2} \cos\left(\frac{k}{2}v\right)\right) \sin v,$$

$$x_3(u, v) = \frac{u}{2} \sin\left(\frac{k}{2}v\right)$$

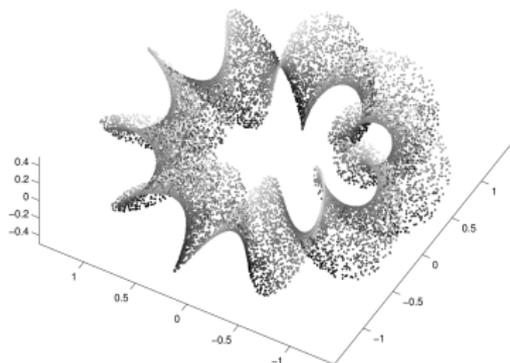


Figure: 16000 points sur le "10-Ruban" de Möbius

n	d_{clust}	d_{levi}	d_{sliv}
100	3	3	3
200	3	3	3
500	3	2	2
1000	3	2	2

Figure: Dimension estimée

Bilan

Résumé

- 1 Algo. 1 : modélisation locale \rightarrow dimension locale, moyennée.

Bilan

Résumé

- 1 Algo. 1 : modélisation locale \rightarrow dimension locale, moyennée.
- 2 Algo. 2 : d minimal pour qu'aucun $d + 1$ -simplexe ne se forme.

Bilan

Résumé

- 1 Algo. 1 : modélisation locale \rightarrow dimension locale, moyennée.
- 2 Algo. 2 : d minimal pour qu'aucun $d + 1$ -simplexe ne se forme.
- 3 Application de l'algorithme de la section précédente

Bilan

Résumé

- 1 Algo. 1 : modélisation locale \rightarrow dimension locale, moyennée.
- 2 Algo. 2 : d minimal pour qu'aucun $d + 1$ -simplexe ne se forme.
- 3 Application de l'algorithme de la section précédente

Contributions

- Détermination automatique d'un paramètre du métamodèle.
- Estimation probabiliste de la dimension basée sur les simplexes.

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes

- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning

- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes

- 4 Applications
 - **Méthodologie**
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Contexte industriel

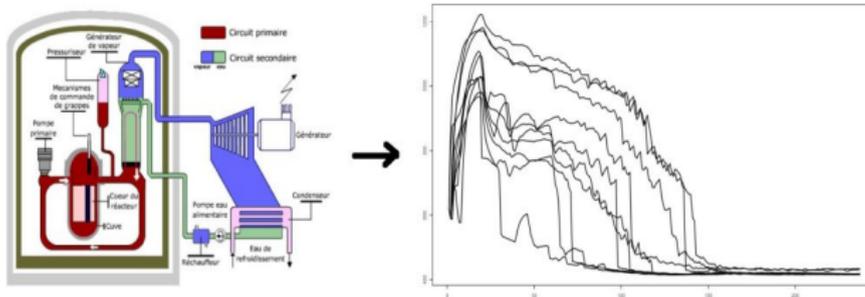
Cadre : durée de vie des cuves.

→ Diverses séquences d'accidents envisagées.

But : estimer leurs probabilités.

Méthodologie

Modélisation → **Simulation** → Calculs.



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

Tests = estimer les erreurs du modèle sur les courbes simulées

Rappel du modèle

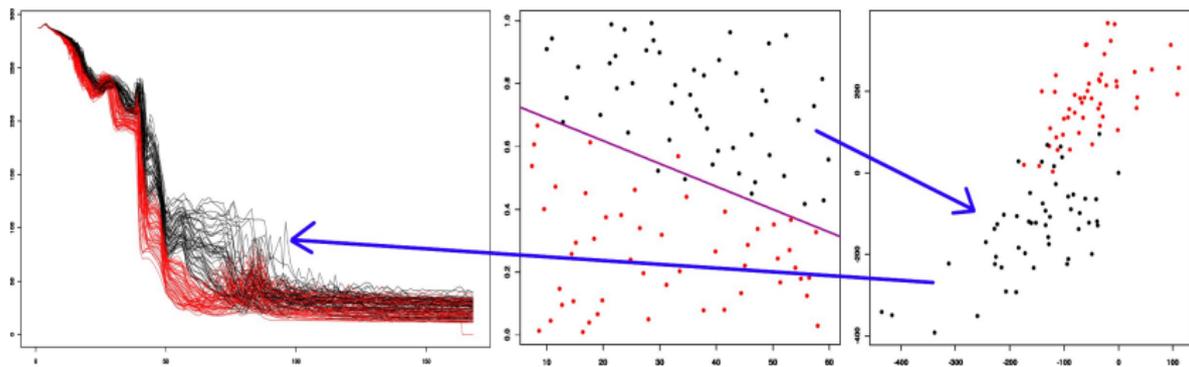


Figure: g. à d. : sorties fonc., entrées 1 - 4, rep. 2D des sorties

- 1 Classification non supervisée des n données en K clusters C_j
+ classification supervisée des entrées x_i .
- 2 Pour chaque cluster C_j ,
 - 1 réduction de la dimension : $r(y_i) = z_i$ représente y_i dans \mathbb{R}^d ;
 - 2 apprentissage d'une fonction de régression : $f(x_i) \simeq z_i$;
 - 3 apprentissage d'une fonction de reconstruction : $R(z_i) \simeq y_i$.

Validation du modèle

Données ($m + m' = n$) :

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

Prédictions du modèle : $\hat{y}'_i = \varphi(x'_i), i = 1, \dots, m'$.

Validation du modèle

Données ($m + m' = n$) :

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

Prédictions du modèle : $\hat{y}'_i = \varphi(x'_i), i = 1, \dots, m'$.

Mesure "absolue" puis relative de l'erreur ponctuelle

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

Validation du modèle

Données ($m + m' = n$) :

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

Prédictions du modèle : $\hat{y}'_i = \varphi(x'_i), i = 1, \dots, m'$.

Mesure "absolue" puis relative de l'erreur ponctuelle

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

$$Q_2[j] = 1 - \frac{m' \cdot MSE[j]}{\sum_{i=1}^{m'} (\bar{y}(j) - y'_i(j))^2} \text{ (comparaison à la moyenne).}$$

$-\infty < Q_2 \leq 1$: $\leq 0 \Rightarrow$ (très) mauvais modèle ;
 $\simeq 1 \Rightarrow$ modèle parfait.

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Test I - courbes de température CATHARE

100 évaluations du modèle,
4 dimensions en entrée,
168 points de discrétisation.

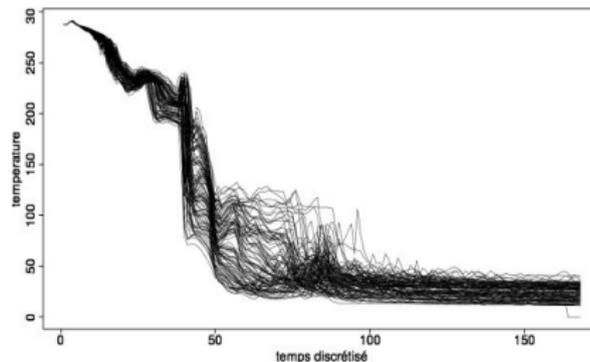


Figure: Les 100 courbes en sortie

Test I - courbes de température CATHARE

100 évaluations du modèle,
4 dimensions en entrée,
168 points de discrétisation.

leave-10-out avec clustering :

MSE à g., Q_2 à d. ; $\hat{d} = 5$

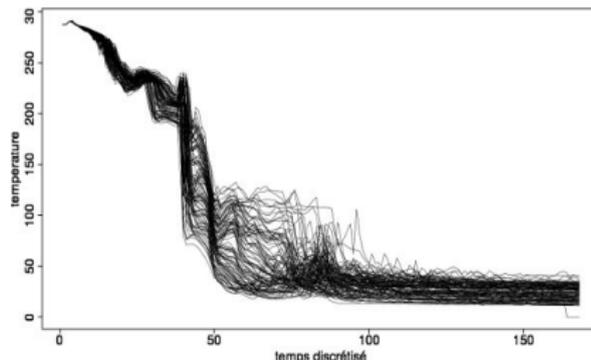


Figure: Les 100 courbes en sortie

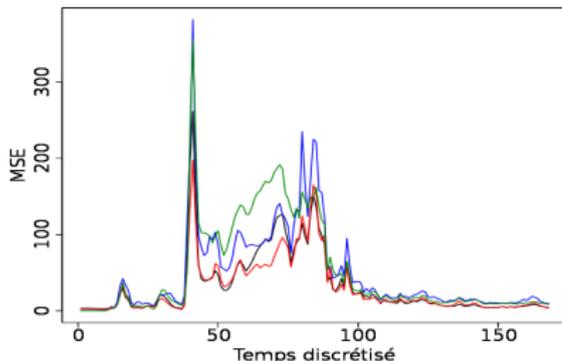
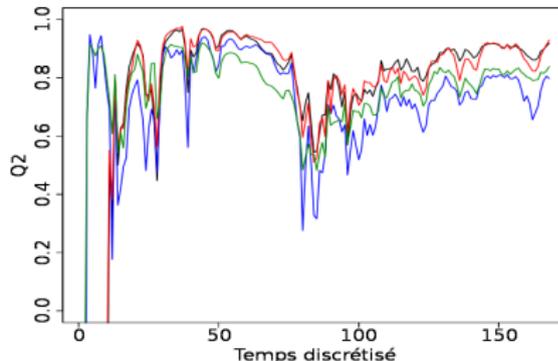


Figure: Noir : ACPF ; bleu : RML ; rouge : ACPFL ; vert : k-PPVF



4 courbes prédites – mauve = courbe cible.

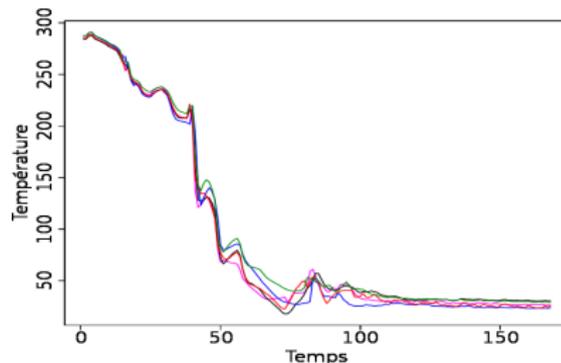
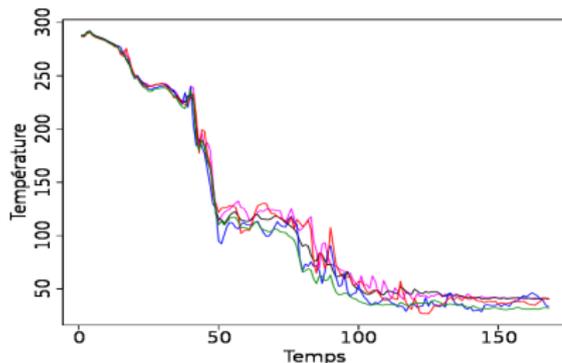


Figure: Noir : ACPF, bleu : RML, rouge : ACPFL, vert : k -PPVF

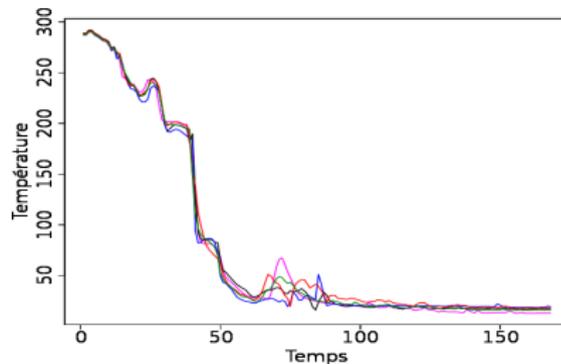
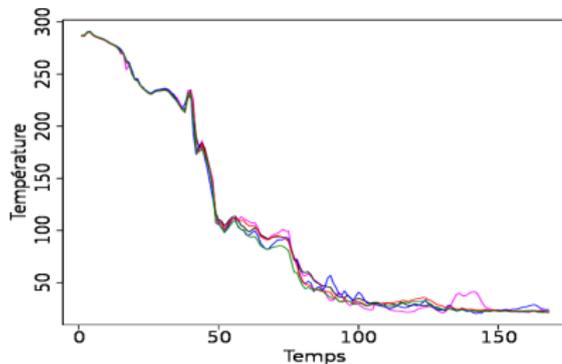


Figure: Noir : ACPF, bleu : RML, rouge : ACPFL, vert : k -PPVF

- 1 Clustering des entrées-sorties
 - Distance
 - Classification ascendante hiérarchique (CAH)
 - Détermination du nombre de classes
- 2 Réduction de la dimension
 - Approches linéaires
 - Riemannian Manifold Learning
- 3 Estimation de la dimension
 - Introduction
 - Quelques algorithmes
- 4 Applications
 - Méthodologie
 - Jeu de données CATHARE I
 - Jeu de données CATHARE II

Test II - courbes de température CATHARE

600 évaluations du modèle,
11 dimensions en entrée,
414 points de discrétisation.

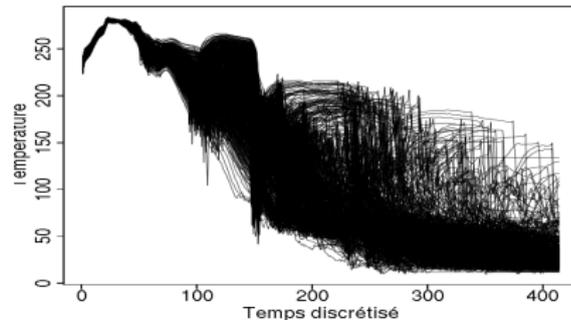


Figure: Les 600 courbes en sortie

Test II - courbes de température CATHARE

600 évaluations du modèle,
11 dimensions en entrée,
414 points de discrétisation.

leave-30-out *sans clustering* :

MSE à g., Q_2 à d. ; $\hat{d} = 8$

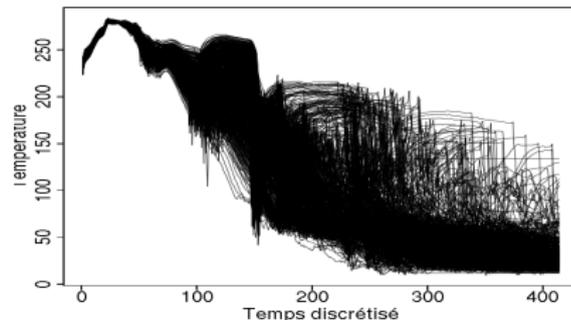


Figure: Les 600 courbes en sortie

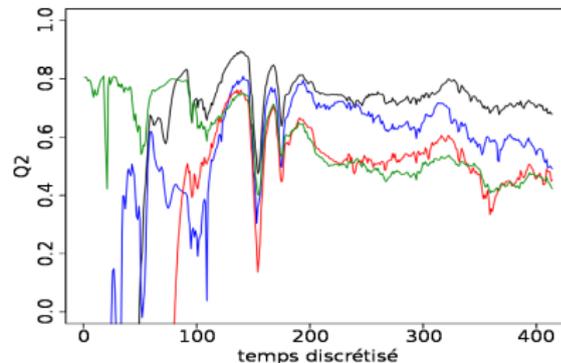
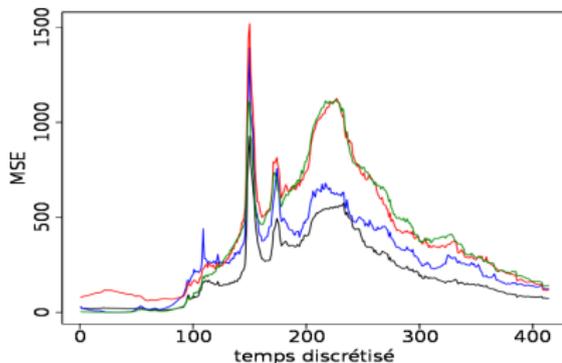


Figure: Noir : ACPF ; bleu : RML ; rouge : ACPFL ; vert : k-PPVF

5 courbes prédites

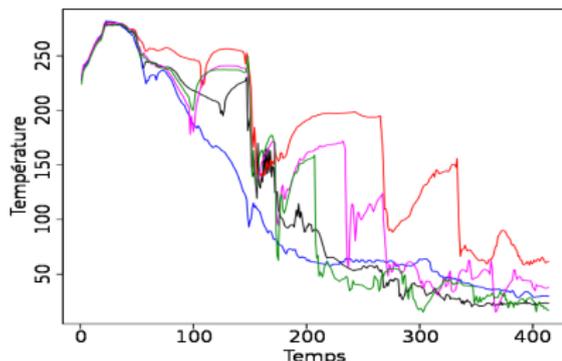


Figure: Courbes cibles

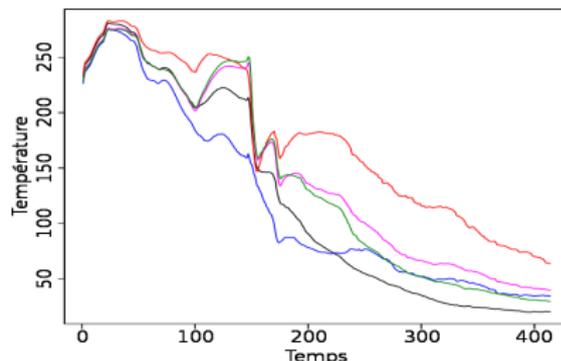


Figure: Preds. ACPF

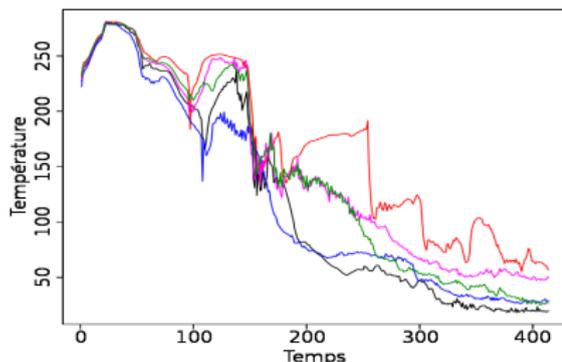


Figure: Preds. RML

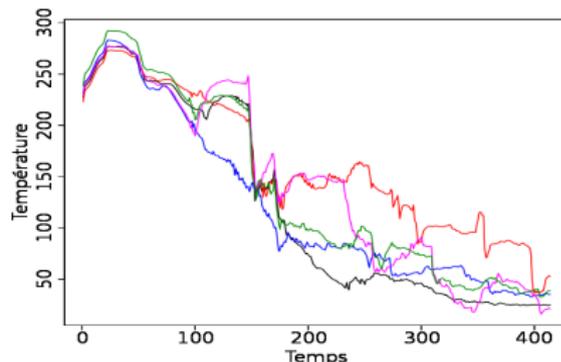


Figure: Preds. ACPFL

Conclusion

Approches linéaire et non linéaire complémentaires.

Conclusion

Approches linéaire et non linéaire complémentaires.

Léger apport du clustering pour CATHARE I.

Conclusion

Approches linéaire et non linéaire complémentaires.

Léger apport du clustering pour CATHARE I.

Bilan des réalisations

Package R métamodèle fonctionnel avec :

- détermination automatique du nombre de clusters ;
- estimation et réduction de la dimension des sorties ;

Conclusion

Approches linéaire et non linéaire complémentaires.

Léger apport du clustering pour CATHARE I.

Bilan des réalisations

Package R métamodèle fonctionnel avec :

- détermination automatique du nombre de clusters ;
- estimation et réduction de la dimension des sorties ;
- méthodes de régression fonctionnelle alternatives ;
- possibilité d'utiliser un mélange de modèles et/ou d'apprendre les résidus fonctionnels ;

Conclusion

Approches linéaire et non linéaire complémentaires.

Léger apport du clustering pour CATHARE I.

Bilan des réalisations

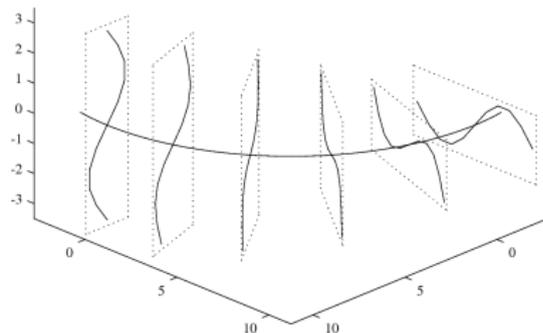
Package R métamodèle fonctionnel avec :

- détermination automatique du nombre de clusters ;
- estimation et réduction de la dimension des sorties ;
- méthodes de régression fonctionnelle alternatives ;
- possibilité d'utiliser un mélange de modèles et/ou d'apprendre les résidus fonctionnels ;
- ajustement heuristique de la plupart des paramètres.

Perspectives

- Opérations préliminaires sur les courbes.
- Distances utilisant sur les dérivées (clustering).
- Courbes principales fonctionnelles (réduction de dimension).

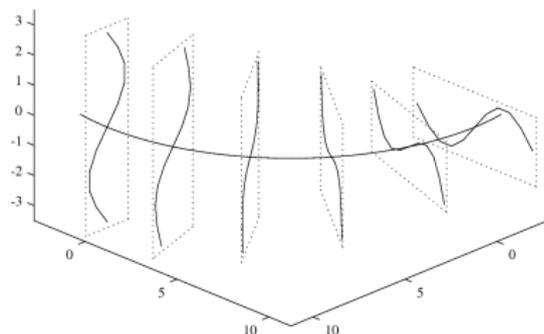
Surface principale en $2D$



Perspectives

- Opérations préliminaires sur les courbes.
- Distances utilisant sur les dérivées (clustering).
- Courbes principales fonctionnelles (réduction de dimension).

Surface principale en $2D$



- Analyse de sensibilité adaptée au métamodèle.
- Prise en compte des corrélations inter-courbes.
- "Recollage" de modèles aux points de "discontinuité".