Statistical methods for multivariate functional data

Steven Golovkine
Supervision: Valentin Patilea[1] and Nicolas Klutchnikoff[2]

[1]CREST, Ensai
[2]IRMAR, Université Rennes 2

June 18$^{\text{th}}$, 2021

GROUPE RENAULT

Table of contents

Automotive safety

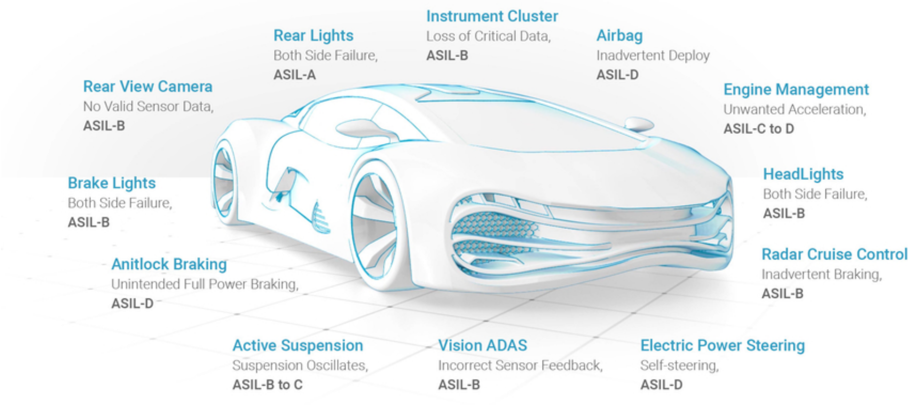


Figure: Automotive Safety Integrity Level for some Advanced Driver-Assistance Systems

Problematic

- Raffaelli et al. [5] propose to combine a statistical approach with simulations to reduce the number of kilometers done physically to validate the reliability of the car. This method is based on test cases generation.
- Cherfi et al. [1] express a need for the identification of driving scenarios that are representatives of the human behavior.
- Our work falls within this industrial context by proposing a method to build a driving scenario database using statistical analysis of real drivings.

Data

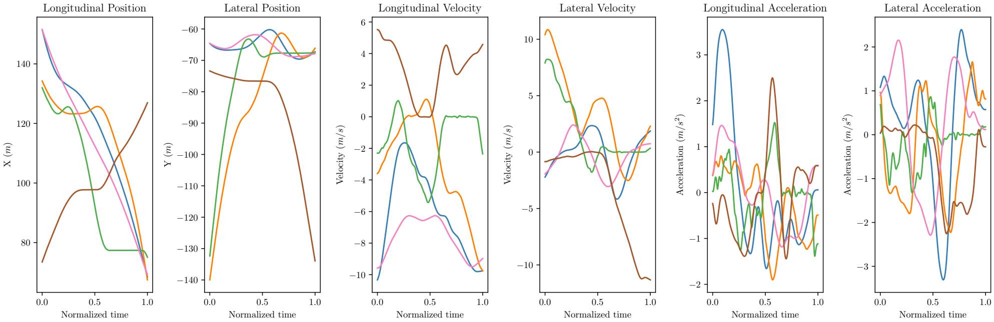
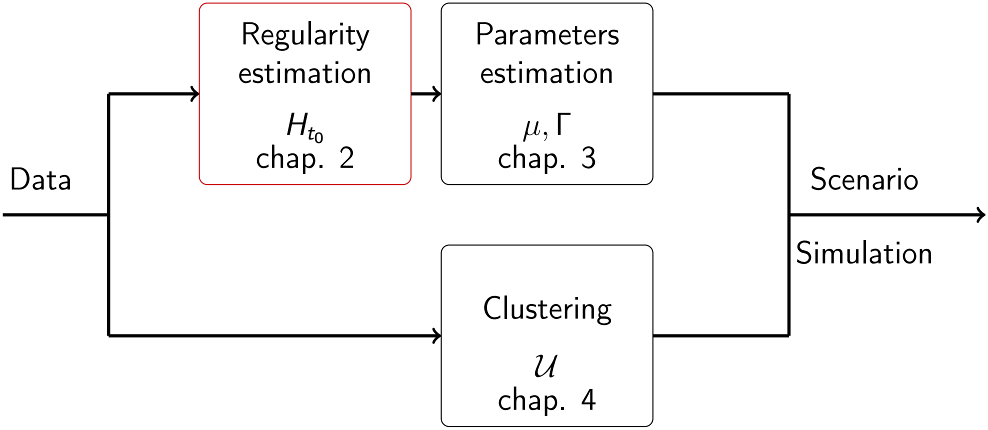


Figure: Sample of trajectories in the rounD dataset.

- We aim to estimate the mean and covariance functions and develop a clustering procedure for these data.

Methodology overview



Theoretical model

- Let
$$I := [0, 1] \quad \text{and} \quad \mathcal{H} := \mathcal{L}^2(I) \times \cdots \times \mathcal{L}^2(I).$$
- We are interested by independent realizations of the P-dimensional stochastic process
$$X = \{(X_1(t_1), \ldots, X_P(t_P)) : t_1, \ldots, t_P \in I\}$$
taking values in $\mathcal{H}$.

Observation model

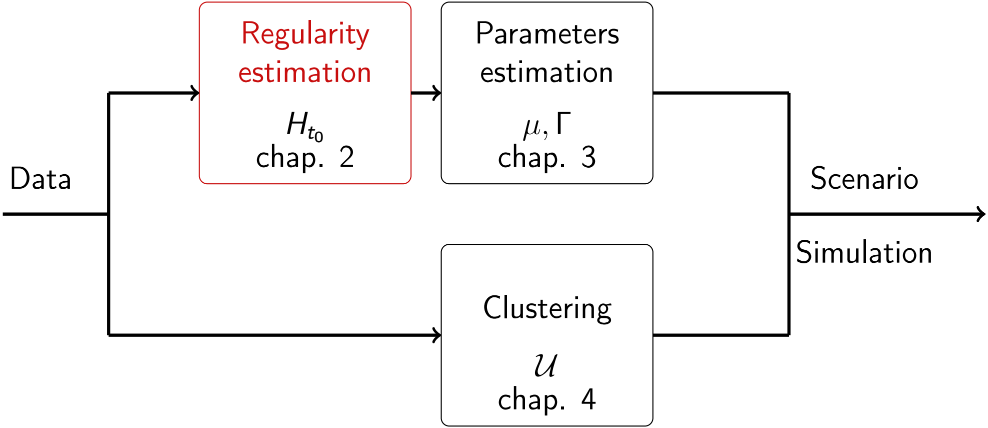
- Let $X^{(n)}, n \in \{1, \ldots, N\}$ be independent trajectories of X.
- In practice, such trajectories cannot be observed at any t.
- Moreover, only noisy data are available:
  - ➥ the observed values on the trajectory $X^{(n)}(\cdot)$ are contaminated with additive errors.
- For any $1 \leq n \leq N$, we observe the random pairs $(T_{\mathrm{m}}^{(n)}, Y_{\mathrm{m}}^{(n)})$ which are defined as:

$$Y_{\mathrm{m}}^{(n)} = X^{(n)}(T_{\mathrm{m}}^{(n)}) + \epsilon_{\mathrm{m}}^{(n)}, \quad \mathrm{m} = (m_1, \ldots, m_P)$$

where

- ➥ $T_{\mathrm{m}}^{(n)} = (T_{m_1}^{(n)}, \ldots, T_{m_P}^{(n)})$ are i.i.d. random sampling points in I;
- ➥ $\epsilon_{\mathrm{m}}^{(n)}$ are i.i.d. random vectors.

Methodology overview



Aim

- The case $P \geq 1$ is build upon the case $P = 1$.
- We aim to
  - ➥ estimate $X^{(n)}(t_0)$ for an arbitrary point $t_0 \in I$ and for each n.
  - ➥ use such estimates to calculate the mean and covariance functions.
- Ideas
  - ➥ Use the large number of curves coming from the same random process
  - ➥ Use the regularity of the sample paths

Aim

- The case $P \geq 1$ is build upon the case $P = 1$.
- We aim to
  - ➥ estimate $X^{(n)}(t_0)$ for an arbitrary point $t_0 \in I$ and for each n.
  - ➥ use such estimates to calculate the mean and covariance functions.
- Ideas
  - ➥ Use the large number of curves coming from the same random process
  - ➥ Use the regularity of the sample paths

Observation model for one component

- For any $1 \leq n \leq N$, we observe the random pairs $(T_m^{(n)}, Y_m^{(n)})$ which are defined as:

$$Y_m^{(n)} = X^{(n)}(T_m^{(n)}) + \epsilon_m^{(n)}, \quad 1 \leq m \leq M_n$$

  where
  - $\epsilon_{\mathrm{m}}^{(n)}$ are i.i.d. random variables with variance σ^2.
- Let $\mathfrak{m} = \mathbb{E}(M_n)$ be the mean number of observations per curve.

Concept of local regularity

- Let $\mathcal{O}_\star$ be a neighborhood of t_0.
- For $H_{t_0} \in (0, 1]$, assume that the stochastic process X satisfies the condition:

$$\mathbb{E}\left((X_u - X_v)^2\right) \asymp L^2 |v - u|^{2H_{t_0}}, \quad \text{with } u \text{ and } v \in \mathcal{O}_\star.$$

- H_{t_0} is called *the local regularity of the process X on $\mathcal{O}_\star$*.
- This quantity is related to the decreasing rate of the ordered eigenvalues of the covariance operator of the process.

$$\text{If, for some } \nu > 1, \lambda_j \sim j^{-\nu},\ j \geq 1, \quad \text{then } 2H = \nu - 1.$$

Estimation: 1st method

- Let K_0 be an integer number and $\mathcal{B} = \{M \geq K_0, T_{(1)} \in \mathcal{O}_\star, \dots, T_{(K_0)} \in \mathcal{O}_\star\}$.
- For k such that $2k - 1 \leq K_0$, let

$$\theta_k = \mathbb{E}\left[(Y_{(2k-1)} - Y_{(k)})^2 1_{\mathcal{B}}\right] = \mathbb{E}\left[(X_{(2k-1)} - X_{(k)})^2 1_{\mathcal{B}}\right] + 2\sigma^2.$$

- Assuming $8k - 7 \leq K_0$, a natural proxy of H_{t_0} is given by

$$H_{t_0}(k) = \frac{\log(\theta_{4k-3} - \theta_{2k-1}) - \log(\theta_{2k-1} - \theta_k)}{2 \log 2}.$$

Estimation: 1st method

- Let K_0 be an integer number and $\mathcal{B} = \{M \geq K_0, T_{(1)} \in \mathcal{O}_\star, \dots, T_{(K_0)} \in \mathcal{O}_\star\}$.
- For k such that $2k - 1 \leq K_0$, let

$$\theta_k = \mathbb{E}\left[(Y_{(2k-1)} - Y_{(k)})^2 1_{\mathcal{B}}\right] = \mathbb{E}\left[(X_{(2k-1)} - X_{(k)})^2 1_{\mathcal{B}}\right] + 2\sigma^2.$$

- Assuming $8k - 7 \leq K_0$, an estimator of H_{t_0} is given by

$$\widehat{H}_{t_0}(k) = \frac{\log(\widehat{\theta}_{4k-3} - \widehat{\theta}_{2k-1}) - \log(\widehat{\theta}_{2k-1} - \widehat{\theta}_k)}{2 \log 2}, \quad \text{if} \quad \widehat{\theta}_{4k-3} > \widehat{\theta}_{2k-1} > \widehat{\theta}_k$$

where

$$\widehat{\theta}_k = \frac{1}{N} \sum_{n=1}^{N} \left(Y_{(2k-1)}^{(n)} - Y_{(k)}^{(n)}\right)^2.$$

Estimation: 2nd method

- For $s, t \in \mathcal{O}_\star$, let

$$\theta(s, t) = \mathbb{E}\left[(X_t - X_s)^2\right] \approx L^2|t - s|^{2H_{t_0}}.$$

- Let t_1 and t_3 be such that $[t_1, t_3] \subset \mathcal{O}_\star$, and denote t_2 the middle point of $[t_1, t_3]$.
- A natural proxy of H_{t_0} is given by

$$\frac{\log(\theta(t_1, t_3)) - \log(\theta(t_1, t_2))}{2\log 2}, \quad \text{if } t_3 - t_1 \text{ is small.}$$

Estimation: 2nd method

- For $s, t \in \mathcal{O}_\star$, let

$$\theta(s,t) = \mathbb{E}\left[(X_t - X_s)^2\right] \approx L^2 |t-s|^{2H_{t_0}}.$$

- Let t_1 and t_3 be such that $[t_1, t_3] \subset \mathcal{O}_\star$, and denote t_2 the middle point of $[t_1, t_3]$.

- An estimator of H_{t_0} is given by

$$\frac{\log(\widehat{\theta}(t_1,t_3)) - \log(\widehat{\theta}(t_1,t_2))}{2\log 2}, \quad \text{if } t_3 - t_1 \text{ is small.}$$

where, given a nonparametric estimator $\widetilde{X}_t$ of X_t,

$$\widehat{\theta}(s,t) = \frac{1}{N}\sum_{n=1}^{N}\left(\widetilde{X_t^{(n)}} - \widetilde{X_s^{(n)}}\right)^2.$$

Concentration

- For any $\epsilon > 0$, we have

$$\mathbb{P}\left(|\widehat{H}_{t_0} - H_{t_0}| > \epsilon\right) \leq \mathfrak{a} \exp\left(-\mathfrak{c} N \mathfrak{D}_{\mathfrak{m}} \epsilon^2\right),$$

where

➥ $\mathfrak{a}$ and $\mathfrak{c}$ are positive constants

➥ $\mathfrak{D}_{\mathfrak{m}}$ is a quantity related to the diameter of $\mathcal{O}_\star$

➥ $\mathfrak{D}_{\mathfrak{m}}$ depends on $\mathfrak{m}$ and such that, for all $a, b > 0$,

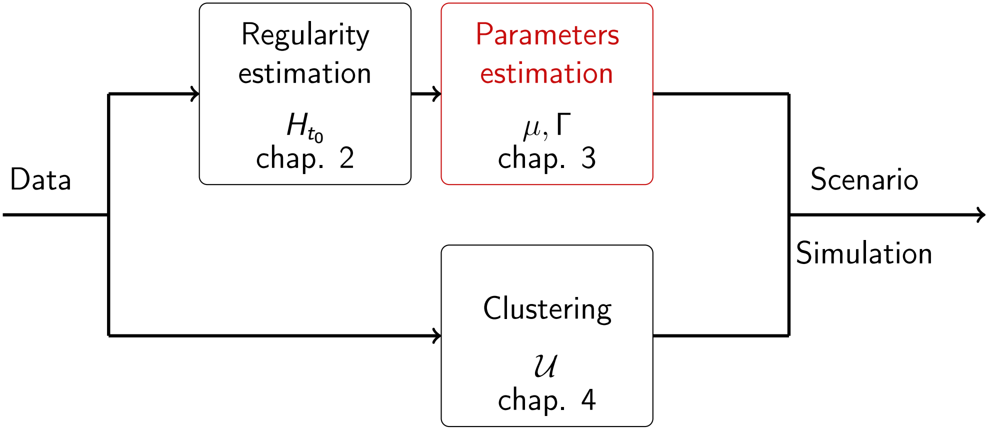
$$\mathfrak{D}_{\mathfrak{m}} \log^a(\mathfrak{m}) \to 0 \quad \text{and} \quad \mathfrak{D}_{\mathfrak{m}} \mathfrak{m}^b \to \infty$$

.

Extension

- Case of regularity larger than 1: the Hölder assumption is made on the d-th derivative
- Case of heteroscedastic noise

Methodology overview



Unfeasible estimators

- If the realizations of the process were observed, the estimators of the mean and covariance functions would be

$$\widetilde{\mu}_N(t) = \frac{1}{N}\sum_{n=1}^{N} X^{(n)}(t), \quad t \in I,$$

$$\widetilde{\Gamma}_N(s,t) = \frac{1}{N-1}\sum_{n=1}^{N}\left(X^{(n)}(s) - \widetilde{\mu}_N(s)\right)\left(X^{(n)}(t) - \widetilde{\mu}_N(t)\right), \quad s, t \in I.$$

Local polynomials

- "Smoothing first, then estimate" paradigm.
- For each $1 \leq n \leq N$, we use a local polynomials approach to build suitable nonparametric estimators $\widehat{X}^{(n)}$ of $X^{(n)}$, defined by

$$\widehat{X}^{(n)}(t, h) = \sum_{m=1}^{M_n} Y_m^{(n)} W_m^{(n)}(t, h),$$

  where $W_m^{(n)}(t, h)$ is a weight function depending on a bandwidth h, which also depends on the regularity H.
- For some curves with sparse observation times, the weights may not be well defined. Let $w_n(t, h)$ be an indicator for non-degenerated smoothed curves at t.

Estimator of the mean function

- For any $t \in I$, let

$$W_N(t, h) = \sum_{n=1}^{N} w_n(t, h).$$

- The estimator of the mean function is

$$\widehat{\mu}_N(t, h) = \frac{1}{W_N(t, h)} \sum_{n=1}^{N} w_n(t, h) \widehat{X}^{(n)}(t), \quad t \in I.$$

- An adaptive optimal bandwidth is

$$\widehat{h}_\mu^\star = C_\mu (N\mathfrak{m})^{-1/(1+2\widehat{H}_t)}.$$

- Note $\widehat{\mu}^\star$ the estimation of the mean using the bandwidth $\widehat{h}_\mu^\star$.

Estimator of the covariance function

- For any $s \neq t$, let

$$W_N(s, t, h) = \sum_{n=1}^{N} w_n(s, h) w_n(t, h).$$

- The estimator of the covariance function is, for $|s - t| > \delta$,

$$\widehat{\Gamma}_N(s, t, h) = \frac{1}{W_N(s, t, h)} \sum_{n=1}^{N} w_n(s, h) \widehat{X}^{(n)}(s) w_n(t, h) \widehat{X}^{(n)}(t) - \widehat{\mu}^\star(s) \widehat{\mu}^\star(t).$$

- An adaptive optimal bandwidth is

$$\widehat{h}_\Gamma^\star = C_\Gamma (N\mathfrak{m})^{-1/(1+2\min(\widehat{H}_s, \widehat{H}_t))}.$$

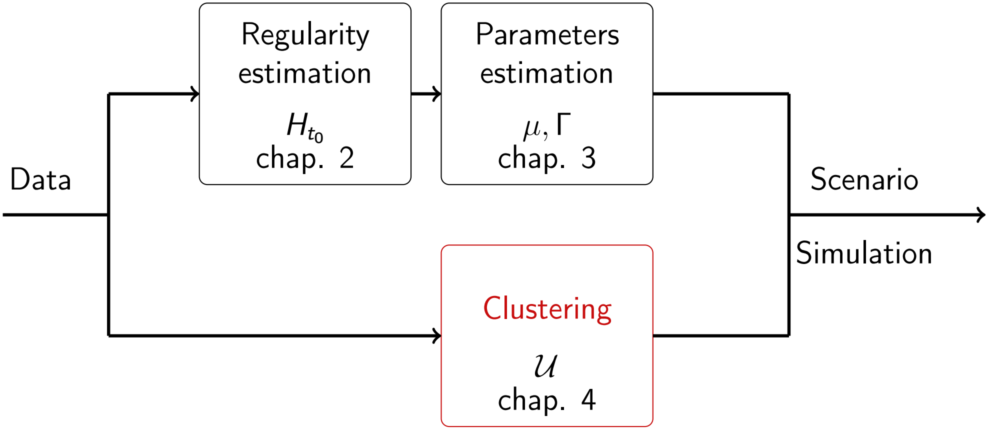
Estimator of the diagonal band of the covariance function

- The previous estimator can be used only outside the diagonal set.
- The covariance function estimator of the diagonal band, for $|s - t| \leq \delta$ is defined as

$$\widehat{\Gamma}_N(s, t, h) = \frac{1}{W_N(u, h)} \sum_{n=1}^{N} w_n(u, h) \widehat{(X^{(n)})^2}(u) - \widehat{\mu}_N^{\star 2}(u, h), \quad u = (s + t)/2$$

$$= \frac{1}{W_N(u, h)} \sum_{n=1}^{N} w_n(u, h) \{Y_m^{(n)}\}^2 W_m^{(n)}(u, h) - \widehat{\mathbb{E}}\left[\sigma^2(u, X_u)\right] - \widehat{\mu}_N^{\star 2}(u)$$

Methodology overview



A mixture model for curves

- Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \dots, K\}$ such that

$$\mathbb{P}(Z = k) = p_k \quad \text{with} \quad p_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1.$$

- We consider that the stochastic process X admits the following decomposition:

$$X(\mathrm{t}) = \sum_{k=1}^{K} \mu_k(\mathrm{t}) 1_{\{Z=k\}} + \sum_{j \geq 1} \xi_j \phi_j(\mathrm{t}), \quad \mathrm{t} \in \mathcal{T},$$

where

  - ➥ $\mu_1, \dots, \mu_K \in \mathcal{H}$ are the mean curves per cluster.
  - ➥ $\{\phi_j\}_{j \geq 1}$ in an orthonormal basis of $\mathcal{H}$ endowed with the inner product $\langle\langle \cdot, \cdot \rangle\rangle$.
  - ➥ For each $1 \leq k \leq K, \xi_j | Z = k \sim \mathcal{N}(0, \sigma_{kj}^2)$, for all $j \geq 1$.

A mixture model for curves

- Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \dots, K\}$ such that

$$\mathbb{P}(Z = k) = p_k \quad \text{with} \quad p_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1.$$

- We consider that the stochastic process X admits the following decomposition:

$$X(\mathrm{t}) = \sum_{k=1}^{K} \mu_k(\mathrm{t}) 1_{\{Z=k\}} + \sum_{j \geq 1} \xi_j \phi_j(\mathrm{t}), \quad \mathrm{t} \in \mathcal{T},$$

where

  - $\mu_1, \dots, \mu_K \in \mathcal{H}$ are the mean curves per cluster.
  - $\{\phi_j\}_{j\geq 1}$ in an orthonormal basis of $\mathcal{H}$ endowed with the inner product $\langle\langle \cdot, \cdot \rangle\rangle$.
  - For each $1 \leq k \leq K, \xi_j | Z = k \sim \mathcal{N}(0, \sigma^2_{kj})$, for all $j \geq 1$.

A mixture model for curves

- Let K be a positive integer, and let Z be a discrete random variable taking values in $\{1, \dots, K\}$ such that

$$\mathbb{P}(Z = k) = p_k \quad \text{with} \quad p_k > 0 \quad \text{and} \quad \sum_{k=1}^{K} p_k = 1.$$

- We consider that the stochastic process X admits the following decomposition:

$$X(\mathrm{t}) = \sum_{k=1}^{K} \mu_k(\mathrm{t}) 1_{\{Z=k\}} + \sum_{j\geq 1} \xi_j \phi_j(\mathrm{t}), \quad \mathrm{t} \in \mathcal{T},$$

where

- ➥ $\mu_1, \dots, \mu_K \in \mathcal{H}$ are the mean curves per cluster.
- ➥ $\{\phi_j\}_{j\geq 1}$ in an orthonormal basis of $\mathcal{H}$ endowed with the inner product $\langle\!\langle \cdot, \cdot \rangle\!\rangle$.
- ➥ For each $1 \leq k \leq K, \xi_j | Z = k \sim \mathcal{N}(0, \sigma_{kj}^2)$, for all $j \geq 1$.

Lemma

Assume X admits the previous decomposition. Let $\{\psi_j\}_{j\geq 1}$ be another orthonormal basis in $\mathcal{H}$ and consider

$$c_j = \langle\langle X - \mu, \psi_j \rangle\rangle, \quad j \geq 1 \quad \text{where} \quad \mu(\cdot) = \sum_{k=1}^{K} p_k \mu_k(\cdot).$$

Then,

$$c_j | Z = k \sim \mathcal{N}(m_{kj}, \tau_{kj}^2),$$

where

$$m_{kj} = \langle\langle \mu_k - \mu, \psi_j \rangle\rangle \quad \text{and} \quad \tau_{kj}^2 = \sum_{l \geq 1} \langle\langle \phi_l, \psi_j \rangle\rangle^2 \sigma_{kl}^2.$$

▶ The clusters will be preserved after expressing the realizations of the process into an orthonormal basis.

fCUBT

- Let $\mathcal{S} = \{X_1, \dots, X_N\}$ be a sample of realizations of the process X.
- We consider the problem of learning a meaningfull partition $\mathcal{U}$ of $\mathcal{S}$.
- For that, the idea is to build a full binary tree using a topdown procedure by recursive splitting.
- The procedure is based on Fraiman et al. [2], adapted to functional data.
- The splitting criterion is similar to the one from Pelleg and Moore [4].

How to split a node?

Given a training sample $\mathcal{S}$ of realizations of X.

i Perform a MFPCA with $\mathtt{n_{comp}}$ components and get the associated eigenvalues and eigenfunctions Φ (see Happ and Greven [3]).

ii Build the matrix C of the projection of the element of $\mathcal{S}$ onto the elements Φ.

iii For each $k = 1, \dots, K_{max}$, fit a k-components GMM using an EM algorithm on the columns of C. The models are denoted by $\{\mathcal{M}_1, \dots, \mathcal{M}_{K_{max}}\}$.

iv Estimate the number of mixture components $\widehat{K}$ as

$$\widehat{K} = \arg \max_{k=1,\dots,K_{max}} \text{BIC}(\mathcal{M}_k).$$

v If $\widehat{K} > 1$, we split the node in two using the model $\mathcal{M}_2$.

How to split a node?

Given a training sample $\mathcal{S}$ of realizations of X.

i Perform a MFPCA with $\mathtt{n}_{\mathtt{comp}}$ components and get the associated eigenvalues and eigenfunctions Φ (see Happ and Greven [3]).

ii Build the matrix C of the projection of the element of $\mathcal{S}$ onto the elements Φ.

iii For each $k = 1, \dots, K_{max}$, fit a k-components GMM using an EM algorithm on the columns of C. The models are denoted by $\{\mathcal{M}_1, \dots, \mathcal{M}_{K_{max}}\}$.

iv Estimate the number of mixture components $\widehat{K}$ as

$$\widehat{K} = \arg \max_{k=1,\dots,K_{max}} \text{BIC}(\mathcal{M}_k).$$

v If $\widehat{K} > 1$, we split the node in two using the model $\mathcal{M}_2$.

How to split a node?

Given a training sample $\mathcal{S}$ of realizations of X.

i Perform a MFPCA with $\mathtt{n_{comp}}$ components and get the associated eigenvalues and eigenfunctions Φ (see Happ and Greven [3]).

ii Build the matrix C of the projection of the element of $\mathcal{S}$ onto the elements Φ.

iii For each $k = 1, \dots, K_{max}$, fit a k-components GMM using an EM algorithm on the columns of C. The models are denoted by $\{\mathcal{M}_1, \dots, \mathcal{M}_{K_{max}}\}$.

iv Estimate the number of mixture components $\widehat{K}$ as

$$\widehat{K} = \arg \max_{k=1,\dots,K_{max}} \text{BIC}(\mathcal{M}_k).$$

v If $\widehat{K} > 1$, we split the node in two using the model $\mathcal{M}_2$.

How to split a node?

Given a training sample $\mathcal{S}$ of realizations of X.

i Perform a MFPCA with $\mathtt{n_{comp}}$ components and get the associated eigenvalues and eigenfunctions Φ (see Happ and Greven [3]).

ii Build the matrix C of the projection of the element of $\mathcal{S}$ onto the elements Φ.

iii For each $k = 1, \dots, K_{max}$, fit a k-components GMM using an EM algorithm on the columns of C. The models are denoted by $\{\mathcal{M}_1, \dots, \mathcal{M}_{K_{max}}\}$.

iv Estimate the number of mixture components $\widehat{K}$ as

$$\widehat{K} = \arg \max_{k=1,\dots,K_{max}} \mathrm{BIC}(\mathcal{M}_k).$$

v If $\widehat{K} > 1$, we split the node in two using the model $\mathcal{M}_2$.

How to split a node?

Given a training sample $\mathcal{S}$ of realizations of X.

i Perform a MFPCA with $\mathtt{n_{comp}}$ components and get the associated eigenvalues and eigenfunctions Φ (see Happ and Greven [3]).

ii Build the matrix C of the projection of the element of $\mathcal{S}$ onto the elements Φ.

iii For each $k = 1, \dots, K_{max}$, fit a k-components GMM using an EM algorithm on the columns of C. The models are denoted by $\{\mathcal{M}_1, \dots, \mathcal{M}_{K_{max}}\}$.

iv Estimate the number of mixture components $\widehat{K}$ as

$$\widehat{K} = \arg \max_{k=1,\dots,K_{max}} \text{BIC}(\mathcal{M}_k).$$

v If $\widehat{K} > 1$, we split the node in two using the model $\mathcal{M}_2$.

- The construction of a branch of the tree is stopped if one of the following criterion is true:
  - The estimation of K is equal to 1
  - There are less than `minsize` elements in the node
- Three hyperparameters have to be set by the user:
  - $\mathtt{n}_{\mathtt{comp}}$ – The number of components to keep for the MFPCA
  - K_{max} – The maximum number of components to consider for the mixture model
  - `minsize` – The minimal number of elements in a node to be considered to be split

Example

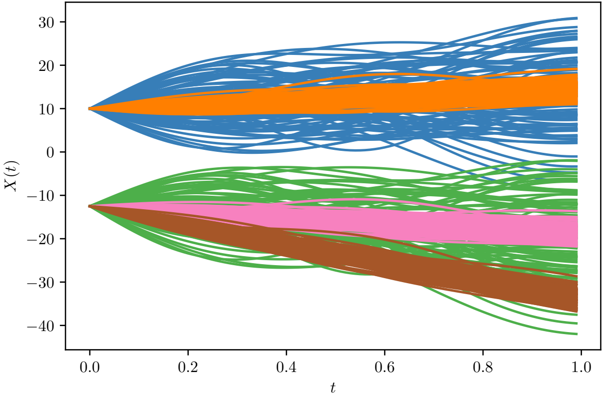


Figure: Example of data

Example of a tree

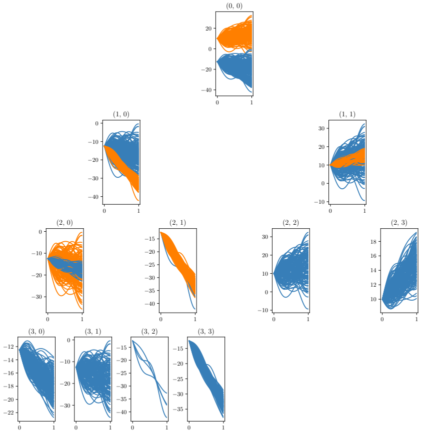
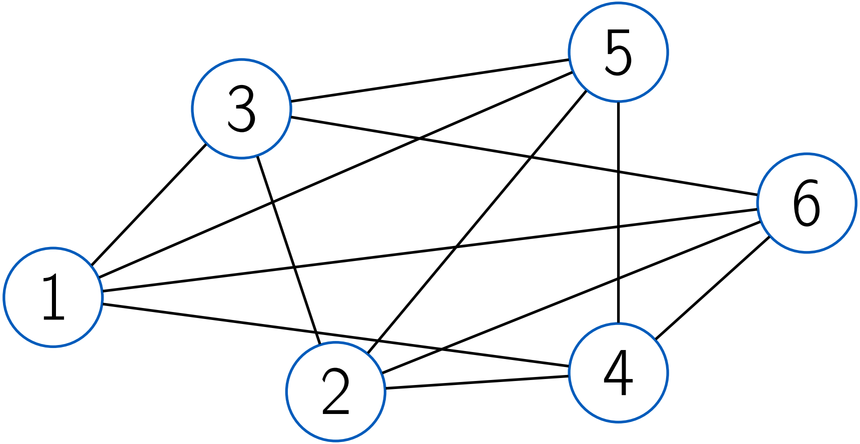
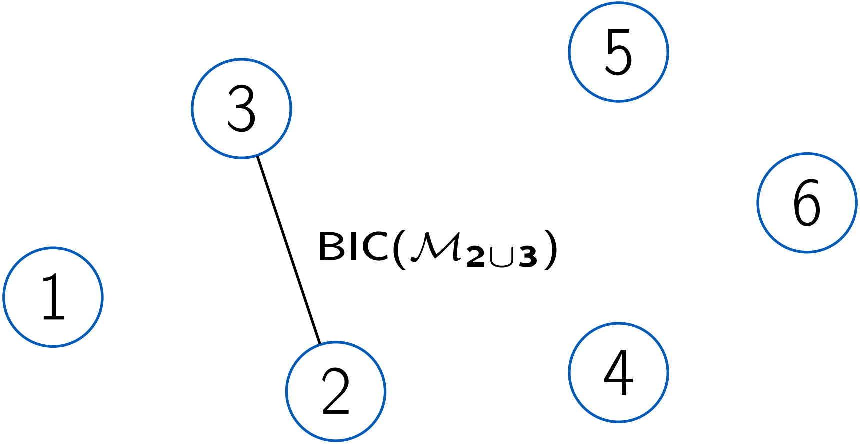


Figure: Example of a grown tree

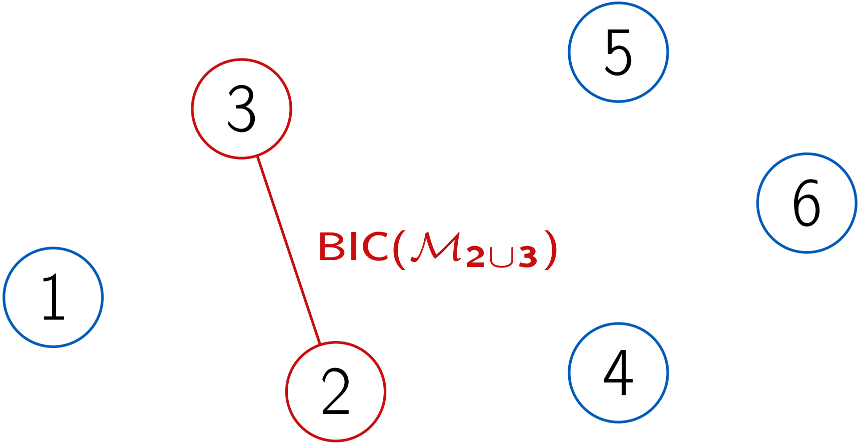
How to join nodes?



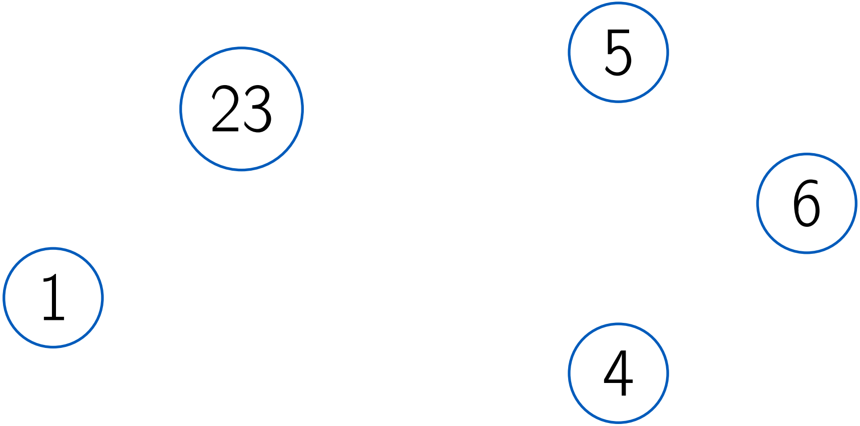
How to join nodes?



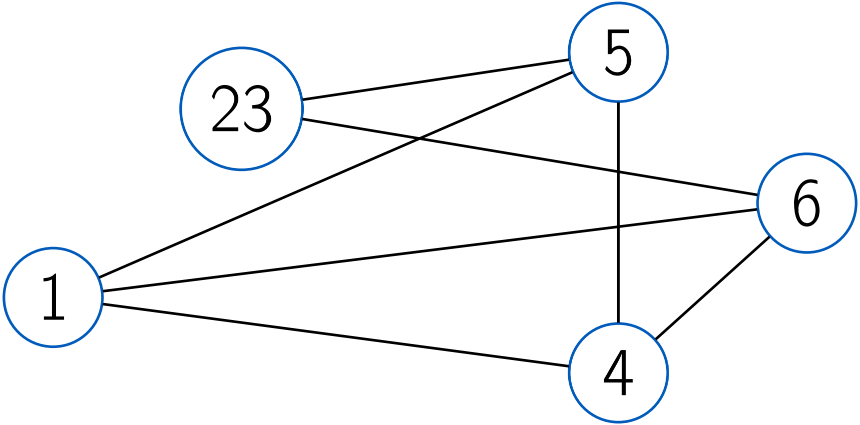
How to join nodes?



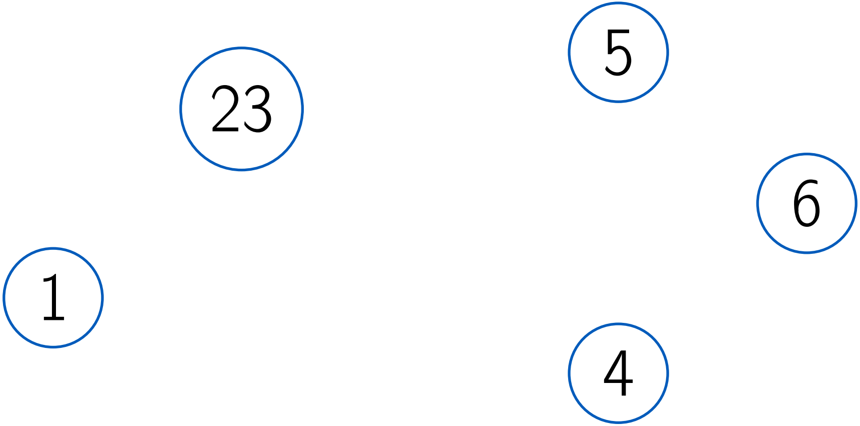
How to join nodes?



How to join nodes?



How to join nodes?



Example: rounD dataset

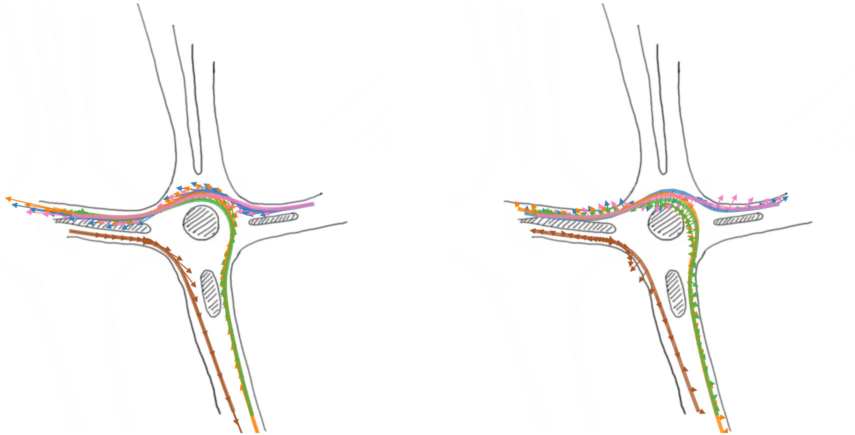


Figure: Sample of trajectories in the rounD dataset.

Example: rounD dataset

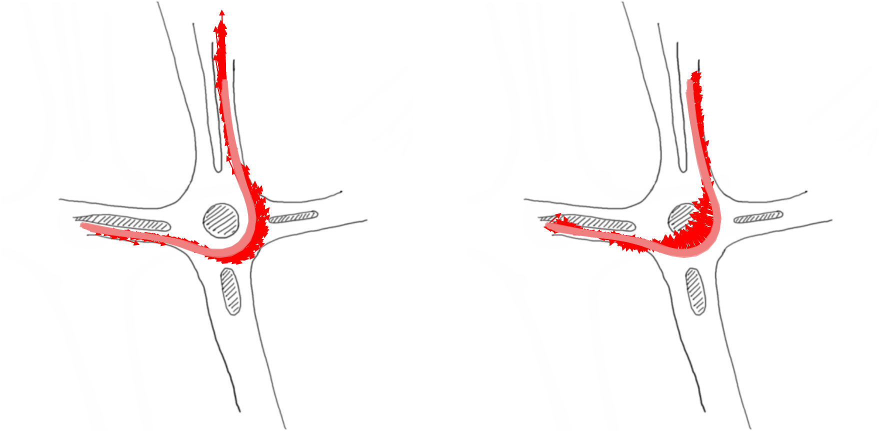


Figure: Example of clusters.

Example: rounD dataset

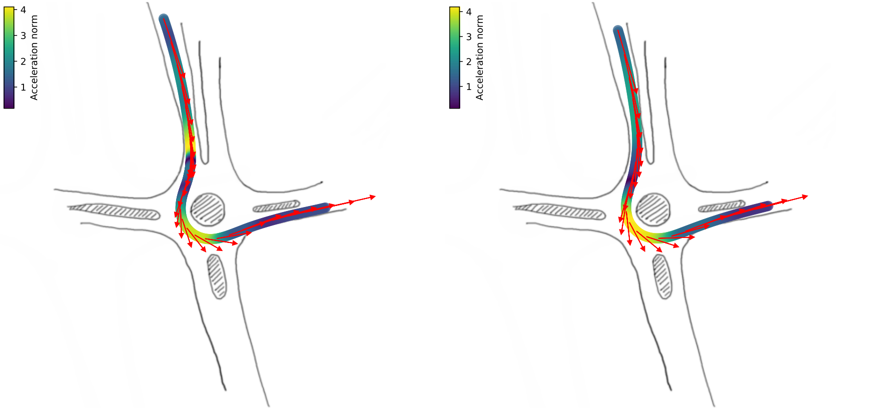


Figure: Two clusters with the same enter/exit.

Conclusion

Regularity estimation for

- performing adaptive optimal smoothing for curve reconstruction
- estimating mean and covariance functions for irregular trajectories

Model-based clustering for a general class of functional data which

- estimates the number of groups
- allows easy predictions
- allows simulations

Propose a methodology to create driving scenarios databases

Methods are implemented in `R` and Python and are available

Extensions

Regularity estimation

- Optimal estimation in functional regression models
- Consider dependent functional data

Clustering

- Integrate wrapping functions to consider phase variation
- Robustify the algorithm

Industrial applications

- Analyze curves data coming from electric vehicles

Contributions

Articles

- Golovkine S., Klutchnikoff N., and Patilea V.. *"Learning the smoothness of noisy curves with application to online curve estimation."* arXiv preprint arXiv:2009.03652 (2020)
- Golovkine S., Klutchnikoff N., and Patilea V.. *"Clustering multivariate functional data using unsupervised binary trees."* arXiv preprint arXiv:2012.05973 (2020)
- Golovkine S.. *"FDApy: a Python package for functional data."* arXiv preprint arXiv:2101.11003 (2021)

Softwares

- FDApy: `https://github.com/StevenGolovkine/FDApy`
- denoisr: `https://github.com/StevenGolovkine/denoisr`

References I

[1] Cherfi, A., Arbaretier, E., and Zhao, L. (2016). Sécurité-innocuité des véhicules autonomes : enjeux et verrous. In *Congrès Lambda Mu 20 de Maîtrise des Risques et de Sûreté de Fonctionnement*. IMdR.

[2] Fraiman, R., Ghattas, B., and Svarc, M. (2010). Clustering using Unsupervised Binary Trees: CUBT. *Computing Research Repository - CORR*.

[3] Happ, C. and Greven, S. (2018). Multivariate Functional Principal Component Analysis for Data Observed on Different (Dimensional) Domains. *Journal of the American Statistical Association*, 113(522):649–659. arXiv: 1509.02029.

[4] Pelleg, D. and Moore, A. (2000). X-means: Extending K-means with Efficient Estimation of the Number of Clusters. In *In Proceedings of the 17th International Conf. on Machine Learning*, pages 727–734. Morgan Kaufmann.

[5] Raffaelli, L., Fayolle, G., and Vallée, F. (2016). ADAS Reliability and Safety. page 10.

Presmoothing estimator

- Let $\Delta_\star = \exp(-\log^{1/2}(\mathfrak{m}))$ be the diameter of $\mathcal{O}_\star$.
- We propose to presmooth the data by local polynomials using the bandwidth:

$$h = \left(\frac{\Delta_\star}{\mathfrak{m}}\right)^{1/3}.$$

Estimation for regularity larger than one

- We test if $\widehat{H}_{t_0} > 1 - \varphi(\mathfrak{m})$ for some decreasing function φ.
- The derivative order is estimated by

$$\widehat{\delta} = \min\{d = 0, \dots, D : \widehat{H}_{t_0} < 1 - \varphi(\mathfrak{m})\}.$$

- The derivatives can be estimated using local polynomials.
- The final estimator of the regularity is $\widehat{\alpha} = \widehat{\delta} + \widehat{H}_{t_0}$.

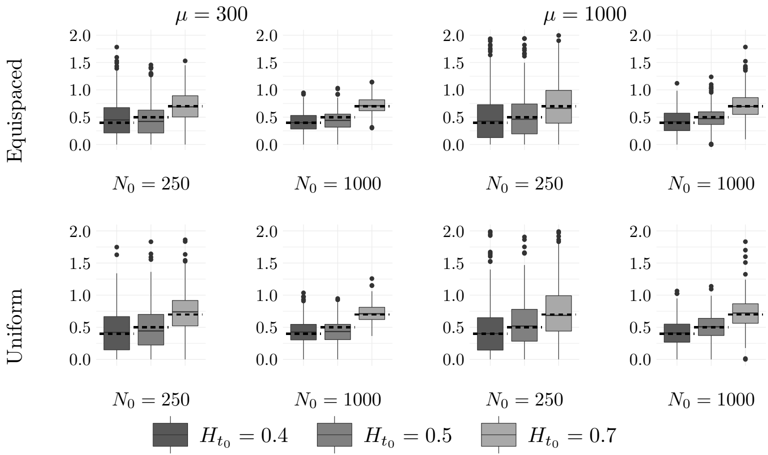


Figure: Estimation of the local regularity for piecewise fractional Brownian motion

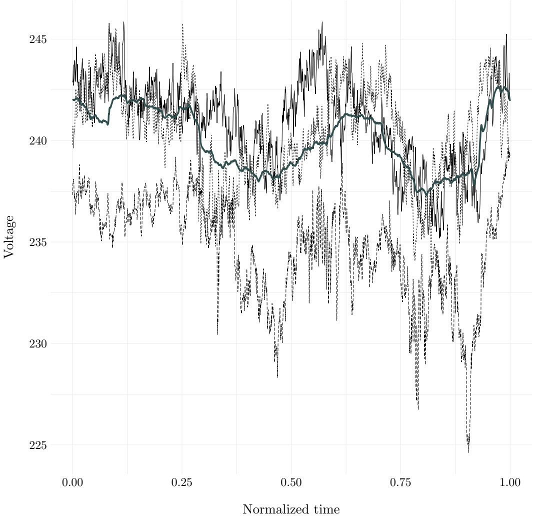


Figure: A sample of curves

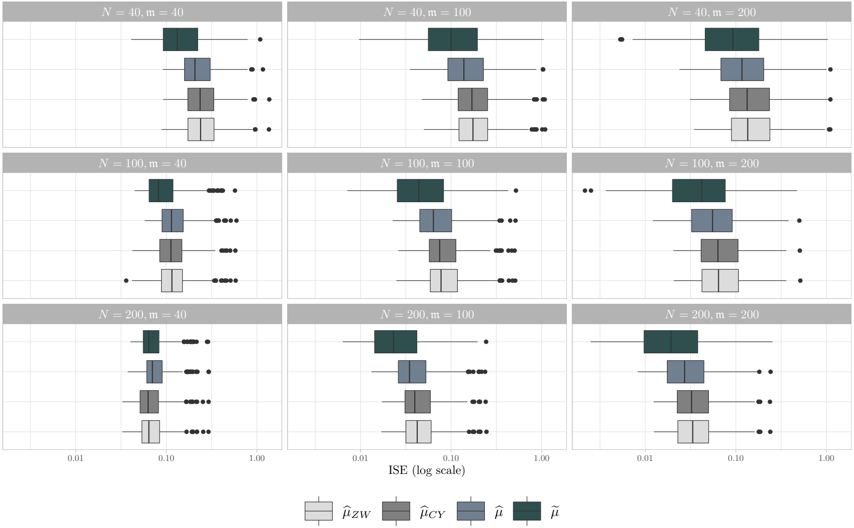


Figure: Estimation of the mean function

Some simulation results

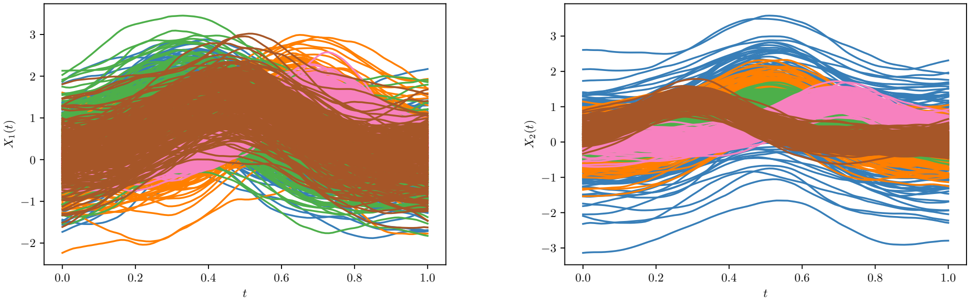


Figure: Simulated data.

Some simulation results: number of clusters

Method	1	2	3	4	5	6	7+
fCUBT	-	-	-	-	0.664	0.238	0.098
Growing	-	-	-	-	0.604	0.182	0.214
FPCA+GMM	-	-	-	-	0.414	0.396	0.19
FunHDDC	0.508	0.492	-	-	-	-	-
Funclust	-	0.066	0.182	0.192	0.200	0.196	0.164
k-means-d_1	-	-	-	-	0.034	0.144	0.822
k-means-d_2	-	0.004	0.01	0.094	0.874	0.010	0.008

Table: Number of clusters

Some simulation results: ARI

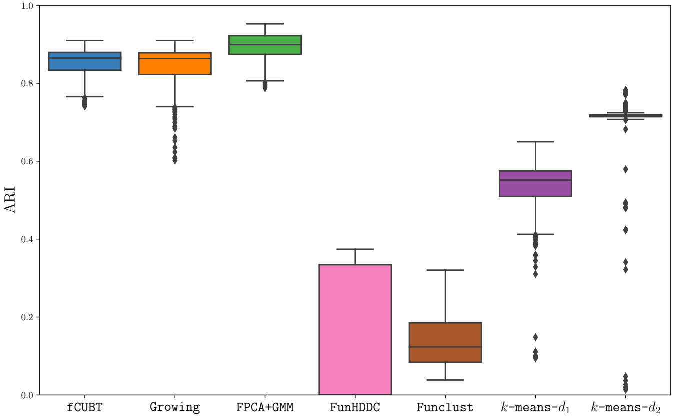


Figure: Rand Index