A study of the manifold hypothesis for functional data by using spectral clustering

Julien Ah-Pine[1,2] **Anne-Françoise Yao**[2]

[1]Univ. of Lyon - ERIC Lab

[2]Univ. of Clermont Auvergne - LMBP

CMStatistics 2019

London, 14th of June 2019

Outline

Outline

Functional Data (FD)

- In many applications, observations are realization of functional data (FD) (curves, time series, signals, images,...).
- Functional Data Analysis (FDA) extends multivariate data analysis techniques to FD or develops specific techniques for FD, see for e.g. [**?**, **?**].
- Objects under study are n real valued functions $\{x_i\}_{i=1,\dots,n}$ in $\mathbb{L}^2([0,T])$, where $T > 0$.
- However $\forall x_i$, we only have p measurements $\{y_{ij}\}_{j=1,\dots,p}$ at discrete time points $\{t_j\}_{j=1,\dots,p}$ in $[0,T]$ and these observations are assumed to be corrupted by noise ϵ_{ij} :

$$y_{ij} = x_i(t_j) + \epsilon_{ij}, \quad \forall i, \forall j$$

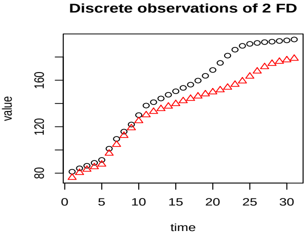
where ϵ_{ij} are assumed to be independent across i and j.

Functional Data Clustering (FDC)

- Given $\{y_{ij}\}_{i,j}$ find a partition of $\{x_i\}_i$ where FD in a class are more similar to each other than to FD in other classes (see for e.g. [**?**]).
- One possible workflow for FDC is the following one :
  - Represent the FD in a **low-dimensional space** using either :
    - Pre-defined finite set of basis functions such as bsplines.
    - Data-driven finite set of basis functions such as truncated Karhunen-Loeve expansion (a.k.a. functional PCA).
  - Apply **multivariate clustering techniques** either :
    - Assuming all FD belong to the whole low-dimensional representation space (e.g. k-means or hierarchical clustering).
    - Assuming that each cluster only belong to a subspace of the representation space (e.g. subspace clustering or model-based functional clustering techniques).

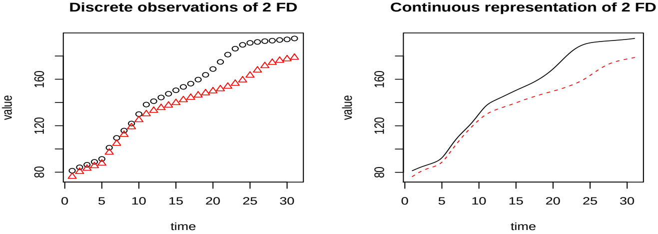
Example : Berkeley Growth data

- $\{y_{ij}\}_{j=1,\dots,p}$ = heights measured at different times t_j.



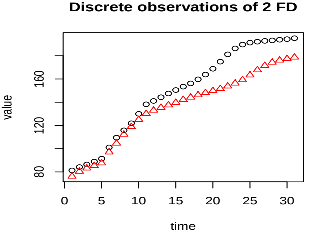
Example : Berkeley Growth data

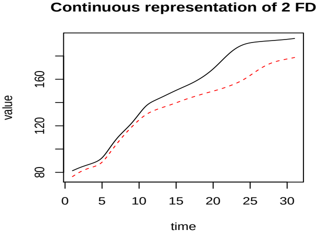
- $\{y_{ij}\}_{j=1,\ldots,p}$ = heights measured at different times t_j.
- x_i= height function of individual i.

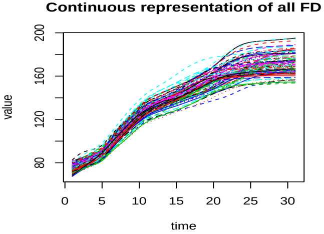


Example : Berkeley Growth data

- $\{y_{ij}\}_{j=1,\dots,p}$ = heights measured at different times t_j.
- x_i= height function of individual i.

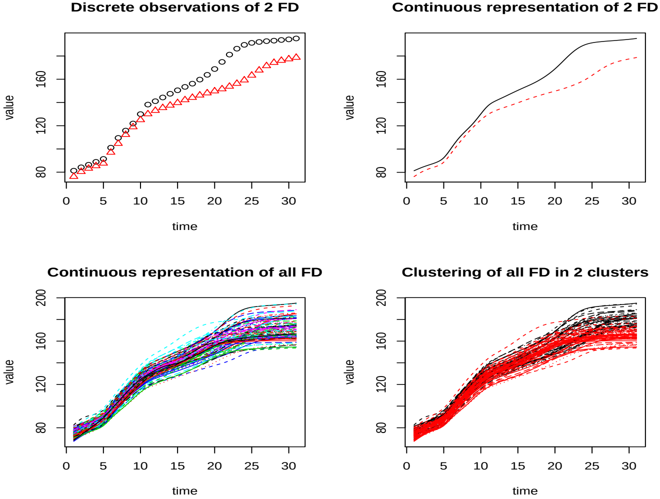






Example : Berkeley Growth data

- $\{y_{ij}\}_{j=1,\dots,p}$ = heights measured at different times t_j.
- x_i= height function of individual i.



Motivations of our study

- Most of previous works do not consider that FD belong to a **RKHS**.

$\rightarrow$ We want to investigate kernel methods for FDC.
E.g. of related work [**?**, **?**].

Motivations of our study

- Most of previous works do not consider that FD belong to a **RKHS**.

$\rightarrow$ We want to investigate kernel methods for FDC.
E.g. of related work [**?**, **?**].

- Most of previous works only use one representation x_i xor Dx_i the derivative functions.

$\rightarrow$ We want to investigate if **information fusion** can leverage the functional nature of the data by considering Sobolev spaces $\mathbb{W}^{1,2}([0, T])$.
E.g. of related work [**?**].

Motivations of our study

- Most of previous works do not consider that FD belong to a **RKHS**.

→ We want to investigate kernel methods for FDC.
E.g. of related work [**?**, **?**].

- Most of previous works only use one representation x_i xor Dx_i the derivative functions.

→ We want to investigate if **information fusion** can leverage the functional nature of the data by considering Sobolev spaces $\mathbb{W}^{1,2}([0, T])$.
E.g. of related work [**?**].

- Most of previous works assume that FD belong to linear spaces or subspaces.

→ We want to investigate further the **manifold hypothesis** : FD belong to low-dimensional non-linear manifold.
E.g. of related work [**?**].

Motivations of our study

- Most of previous works do not consider that FD belong to a **RKHS**.

$\rightarrow$ We want to investigate kernel methods for FDC.
E.g. of related work [**?**, **?**].

- Most of previous works only use one representation x_i xor Dx_i the derivative functions.

$\rightarrow$ We want to investigate if **information fusion** can leverage the functional nature of the data by considering Sobolev spaces $\mathbb{W}^{1,2}([0, T])$.
E.g. of related work [**?**].

- Most of previous works assume that FD belong to linear spaces or subspaces.

$\rightarrow$ We want to investigate further the **manifold hypothesis** : FD belong to low-dimensional non-linear manifold.
E.g. of related work [**?**].

$\Rightarrow$ We investigate these points **jointly** and from an **empirical viewpoint** using **20 benchmarks** and by using **spectral clustering (SC)**.

Outline

Spectral clustering (SC) in a nutshell

- Methods developed in the ML community since the early 2000's.
- Capture the intrinsic geometry of the data.
- Similarity, neighbor end Laplacian graphs are important concepts.
- Methodology : **use the spectral decomposition of the Laplacian matrix as an embedding** of the graph nodes in an Euclidean space then partition the nodes using k-means.
- Motivations : the eigenvalues and eigenvectors of the Laplacian encode information about the connected components (and more generally clusters) of the graph, they also provide solutions to (relaxed) graph cuts problems.
- See for e.g. [**?**] for an introduction.

Similarity and Neighbor graphs

- Similarities between objects as a **weighted undirected graph** $G = (\mathbb{V}, \mathbb{E})$:
  - $\mathbb{V} = \{x_1, \ldots, x_n\}$ is the set of nodes : objects to cluster.
  - $\mathbb{E}$ is the set of edges : pairs of objects that are similar to each other.
- Edges are weighted : if $(x_i, x_j) \in \mathbb{E}$ then $K(x_i, x_j) > 0$ is the **measure of the similarity**.
- G is represented by a **weighted adjacency matrix** denoted $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,n}$ with :

$$w_{ij} = \begin{cases} K(x_i, x_j) & \text{if } (x_i, x_j) \in \mathbb{E} \\ 0 & \text{else} \end{cases}$$

Similarity and Neighbor graphs

- Similarities between objects as a **weighted undirected graph** $G = (\mathbb{V}, \mathbb{E})$:
  - $\mathbb{V} = \{x_1, \dots, x_n\}$ is the set of nodes : objects to cluster.
  - $\mathbb{E}$ is the set of edges : pairs of objects that are similar to each other.
- Edges are weighted : if $(x_i, x_j) \in \mathbb{E}$ then $K(x_i, x_j) > 0$ is the **measure of the similarity**.
- G is represented by a **weighted adjacency matrix** denoted $\mathbf{W} = (w_{ij})_{i,j=1,\dots,n}$ with :

$$w_{ij} = \begin{cases} K(x_i, x_j) & \text{if } (x_i, x_j) \in \mathbb{E} \\ 0 & \text{else} \end{cases}$$

- K is a **kernel function** : objects belong to an RKHS.

Similarity and Neighbor graphs

- Similarities between objects as a **weighted undirected graph** $G = (\mathbb{V}, \mathbb{E})$:
  - $\mathbb{V} = \{x_1, \ldots, x_n\}$ is the set of nodes : objects to cluster.
  - $\mathbb{E}$ is the set of edges : pairs of objects that are similar to each other.
- Edges are weighted : if $(x_i, x_j) \in \mathbb{E}$ then $K(x_i, x_j) > 0$ is the **measure of the similarity**.
- G is represented by a **weighted adjacency matrix** denoted $\mathbf{W} = (w_{ij})_{i,j=1,\ldots,n}$ with :

$$w_{ij} = \begin{cases} K(x_i, x_j) & \text{if } (x_i, x_j) \in \mathbb{E} \\ 0 & \text{else} \end{cases}$$

- K is a **kernel function** : objects belong to an RKHS.
- We can sparsify $\mathbf{W}$ and have a k **nearest neighbor graph** in order to strengthen the manifold hypothesis.

Laplacian matrix and its normalization

- Let $\mathbf{D} = (d_{ij})_{i,j=1,\dots,n}$ be the **degree matrix** defined by :

$$d_{ij} = \begin{cases} d_i & \text{if } i = j \\ 0 & \text{else} \end{cases}$$

  with $d_i = \sum_{j=1}^{n} w_{ij}$, $\forall i = 1, \dots, n$.

- The **Laplacian matrix** of G denoted $\mathbf{L}$ is given by :

$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

- Its **(symmetric) normalization** denoted $\mathbf{L}_{sym}$ is defined by :

$$\begin{aligned} \mathbf{L}_{sym} &= \mathbf{D}^{-1/2}\mathbf{L}\mathbf{D}^{-1/2} \\ &= \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} \end{aligned}$$

  with $\mathbf{I}$ the identity matrix of order n.

Properties of the normalized Laplacian matrix

Property.

- $\mathbf{L}_{sym}$ *can be viewed as a quadratic form (that we aim at minimizing) :*

$$\mathbf{f}^\top \mathbf{L}_{sym}\mathbf{f} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}\left(\frac{f_i}{\sqrt{d_i}} - \frac{f_j}{\sqrt{d_j}}\right)^2, \forall \mathbf{f} \in \mathbb{R}^n$$

- $\mathbf{L}_{sym}$ *is* **symmetric and psd** *:*

$$0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$$

- *The multiplicity order k of the* **null eigenvalue** *is the number of* **connected components** *of G. Let denote the latter subset of nodes as $C_1, \ldots, C_k$. The eigen subspace associated to λ_1 is spanned by* $\mathbf{D}^{1/2}\mathbf{1}_{C_1}, \ldots, \mathbf{D}^{1/2}\mathbf{1}_{C_k}$ *where $\mathbf{1}_{C_l}$ is the assignment vector of C_l.*

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$

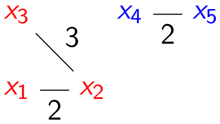


Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$

$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}$$

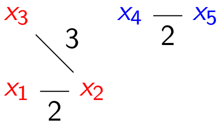
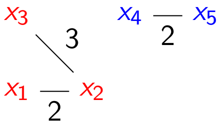


Illustration with a disconnected graph

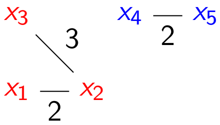
- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$



$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \qquad \rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$

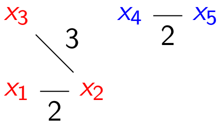


$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \qquad \rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\rightarrow \mathbf{L} = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -2 & 2 \end{pmatrix}$$

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$

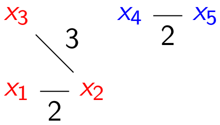


$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \qquad \rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\rightarrow \mathbf{L} = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -2 & 2 \end{pmatrix} \qquad \rightarrow \mathbf{L}_{sym} = \begin{pmatrix} 1 & -.63 & 0 & 0 & 0 \\ -.63 & 1 & -.77 & 0 & 0 \\ 0 & -.77 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$



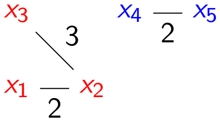
$$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix} \qquad \rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$$

$$\rightarrow \mathbf{L} = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -2 & 2 \end{pmatrix} \qquad \rightarrow \mathbf{L}_{sym} = \begin{pmatrix} 1 & -.63 & 0 & 0 & 0 \\ -.63 & 1 & -.77 & 0 & 0 \\ 0 & -.77 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

$\rightarrow$ Spectra of $\mathbf{L}_{sym}$: $\{2, 2, 1, 0, 0\}$

Illustration with a disconnected graph

- $\mathbb{V} = \{x_1, x_2, x_3, x_4, x_5\}$
- $\mathbb{E} = \{\underbrace{(x_1, x_2)}_{2}, \underbrace{(x_2, x_3)}_{3}, \underbrace{(x_4, x_5)}_{2}\}$



$\rightarrow \mathbf{W} = \begin{pmatrix} 0 & 2 & 0 & 0 & 0 \\ 2 & 0 & 3 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}$ $\rightarrow \mathbf{D} = \begin{pmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 5 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 2 \end{pmatrix}$

$\rightarrow \mathbf{L} = \begin{pmatrix} 2 & -2 & 0 & 0 & 0 \\ -2 & 5 & -3 & 0 & 0 \\ 0 & -3 & 3 & 0 & 0 \\ 0 & 0 & 0 & 2 & -2 \\ 0 & 0 & 0 & -2 & 2 \end{pmatrix}$ $\rightarrow \mathbf{L}_{sym} = \begin{pmatrix} 1 & -.63 & 0 & 0 & 0 \\ -.63 & 1 & -.77 & 0 & 0 \\ 0 & -.77 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$

$\rightarrow$ Spectra of $\mathbf{L}_{sym}$: $\{2,2,1,0,0\}$ $\rightarrow \underbrace{\mathbf{D}^{1/2}\mathbf{1}_{C_1}}_{\mathbf{f}_0^1} = \begin{pmatrix} 1.41 \\ 2.24 \\ 1.73 \\ 0 \\ 0 \end{pmatrix}$ and $\underbrace{\mathbf{D}^{1/2}\mathbf{1}_{C_2}}_{\mathbf{f}_0^2} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1.41 \\ 1.41 \end{pmatrix}$.

Outline

Workflow

1. **Smoothing** : from $\{y_{ij}\}_{i,j}$ to $\{x_i\}_i$:
   - Basis functions are cubic bspline $\{\phi_k\}_{k=1,\dots,q}$ with $q = 4 + p$:

$$x_i(t) = \mathbf{c}_i^\top \phi(t) = \sum_{k=1}^{q} c_{ik}\phi_k(t)$$

     where $\mathbf{c}_i = \begin{pmatrix} c_{i1} & \dots & c_{iq} \end{pmatrix}^\top$ and $\phi(t) = \begin{pmatrix} \phi_1(t) & \dots & \phi_q(t) \end{pmatrix}^\top \in \mathbb{R}^q$.
   - We find $\mathbf{c}_i$ as follows :

$$\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^q} \sum_{j=1}^{p} (y_{ij} - x_i(t_j))^2 + \lambda \int_0^T D^2 x_i(t)dt$$

     where D is the differential operator and λ is the smoothing coefficient selected in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ wrt the GCV criterion.

Workflow

1. **Smoothing** : from $\{y_{ij}\}_{i,j}$ to $\{x_i\}_i$:
   - Basis functions are cubic bspline $\{\phi_k\}_{k=1,\dots,q}$ with $q = 4 + p$:

$$x_i(t) = \mathbf{c}_i^\top \phi(t) = \sum_{k=1}^{q} c_{ik}\phi_k(t)$$

   where $\mathbf{c}_i = \begin{pmatrix} c_{i1} & \dots & c_{iq} \end{pmatrix}^\top$ and $\phi(t) = \begin{pmatrix} \phi_1(t) & \dots & \phi_q(t) \end{pmatrix}^\top \in \mathbb{R}^q$.
   - We find $\mathbf{c}_i$ as follows :

$$\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^q} \sum_{j=1}^{p} (y_{ij} - x_i(t_j))^2 + \lambda \int_0^T D^2 x_i(t)dt$$

   where D is the differential operator and λ is the smoothing coefficient selected in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^0\}$ wrt the GCV criterion.
2. **Center** the $\{x_i\}_i$ and compute **derivatives** $\{Dx_i\}_i$.

Workflow

1. **Smoothing** : from $\{y_{ij}\}_{i,j}$ to $\{x_i\}_i$:
   - Basis functions are cubic bspline $\{\phi_k\}_{k=1,\dots,q}$ with $q = 4 + p$:

     $$x_i(t) = \mathbf{c}_i^\top \phi(t) = \sum_{k=1}^{q} c_{ik}\phi_k(t)$$

     where $\mathbf{c}_i = \begin{pmatrix} c_{i1} & \dots & c_{iq} \end{pmatrix}^\top$ and $\phi(t) = \begin{pmatrix} \phi_1(t) & \dots & \phi_q(t) \end{pmatrix}^\top \in \mathbb{R}^q$.
   - We find $\mathbf{c}_i$ as follows :

     $$\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^q} \sum_{j=1}^{p}(y_{ij} - x_i(t_j))^2 + \lambda \int_0^T D^2 x_i(t)dt$$

     where D is the differential operator and λ is the smoothing coefficient selected in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ wrt the GCV criterion.
2. **Center** the $\{x_i\}_i$ and compute **derivatives** $\{Dx_i\}_i$.
3. Compute the **Gram matrix S** wrt a given kernel function.

Workflow

1. **Smoothing** : from $\{y_{ij}\}_{i,j}$ to $\{x_i\}_i$:
   - Basis functions are cubic bspline $\{\phi_k\}_{k=1,\dots,q}$ with $q = 4 + p$:

     $$x_i(t) = \mathbf{c}_i^\top \phi(t) = \sum_{k=1}^{q} c_{ik}\phi_k(t)$$

     where $\mathbf{c}_i = \begin{pmatrix} c_{i1} & \dots & c_{iq} \end{pmatrix}^\top$ and $\phi(t) = \begin{pmatrix} \phi_1(t) & \dots & \phi_q(t) \end{pmatrix}^\top \in \mathbb{R}^q$.
   - We find $\mathbf{c}_i$ as follows :

     $$\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^q} \sum_{j=1}^{p} (y_{ij} - x_i(t_j))^2 + \lambda \int_0^T D^2 x_i(t) dt$$

     where D is the differential operator and λ is the smoothing coefficient selected in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ wrt the GCV criterion.
2. **Center** the $\{x_i\}_i$ and compute **derivatives** $\{Dx_i\}_i$.
3. Compute the **Gram matrix S** wrt a given kernel function.
4. Perform **clustering procedures**.

Workflow

1. **Smoothing** : from $\{y_{ij}\}_{i,j}$ to $\{x_i\}_i$:
   - Basis functions are cubic bspline $\{\phi_k\}_{k=1,\dots,q}$ with $q = 4 + p$:

   $$x_i(t) = \mathbf{c}_i^\top \phi(t) = \sum_{k=1}^{q} c_{ik}\phi_k(t)$$

   where $\mathbf{c}_i = \begin{pmatrix} c_{i1} & \dots & c_{iq} \end{pmatrix}^\top$ and $\phi(t) = \begin{pmatrix} \phi_1(t) & \dots & \phi_q(t) \end{pmatrix}^\top \in \mathbb{R}^q$.
   - We find $\mathbf{c}_i$ as follows :

   $$\mathbf{c}_i = \arg\min_{\mathbf{c}\in\mathbb{R}^q} \sum_{j=1}^{p} (y_{ij} - x_i(t_j))^2 + \lambda \int_0^T D^2 x_i(t)dt$$

   where D is the differential operator and λ is the smoothing coefficient selected in $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}\}$ wrt the GCV criterion.
2. **Center** the $\{x_i\}_i$ and compute **derivatives** $\{Dx_i\}_i$.
3. Compute the **Gram matrix S** wrt a given kernel function.
4. Perform **clustering procedures**.
5. **Evaluate** clustering outputs and **compare** the results.

Kernel/Representation/Sparsification

- FD are centered : $\sum_{i=1}^{n} x_i(t) = 0, \forall t \in [0, T]$.

Kernel/Representation/Sparsification

- FD are centered : $\sum_{i=1}^{n} x_i(t) = 0, \forall t \in [0, T]$.
- Different **Hilbert spaces** :
  - 00 $x_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2}$.
  - 11 $Dx_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
  - 01 $x_i \in \mathbb{W}^{1,2}([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} + \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.

Kernel/Representation/Sparsification

- FD are centered : $\sum_{i=1}^{n} x_i(t) = 0, \forall t \in [0, T]$.
- Different **Hilbert spaces** :
  - 00 $x_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2}$.
  - 11 $Dx_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
  - 01 $x_i \in \mathbb{W}^{1,2}([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} + \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
- Different **kernel functions (RKHS)** :
  - Linear kernel : $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} = \int_0^T x_i(t)x_j(t)dt$
  - Gaussian Kernel [?] : $K_g(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2_{\mathbb{L}^2}}{\sigma_i \sigma_j}\right)$

Kernel/Representation/Sparsification

- FD are centered : $\sum_{i=1}^{n} x_i(t) = 0, \forall t \in [0, T]$.
- Different **Hilbert spaces** :
  - 00 $x_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2}$.
  - 11 $Dx_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
  - 01 $x_i \in \mathbb{W}^{1,2}([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} + \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
- Different **kernel functions (RKHS)** :
  - Linear kernel : $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} = \int_0^T x_i(t) x_j(t) dt$
  - Gaussian Kernel [?] : $K_g(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2_{\mathbb{L}^2}}{\sigma_i \sigma_j}\right)$
- Different **sparsifications** :
  - 0 "Connected" graph : $w_{ij} = max(K(x_i, x_j), 0)$.
  - 1 k nearest-neighbor graph (with k=7).

Kernel/Representation/Sparsification

- FD are centered : $\sum_{i=1}^{n} x_i(t) = 0, \forall t \in [0, T]$.
- Different **Hilbert spaces** :
  - 00 $x_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2}$.
  - 11 $Dx_i \in \mathbb{L}^2([0, T])$, e.g. $K_l(x_i, x_j) = \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
  - 01 $x_i \in \mathbb{W}^{1,2}([0, T])$, e.g. $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} + \langle Dx_i, Dx_j \rangle_{\mathbb{L}^2}$.
- Different **kernel functions (RKHS)** :
  - Linear kernel : $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2} = \int_0^T x_i(t)x_j(t)dt$
  - Gaussian Kernel [?] : $K_g(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2_{\mathbb{L}^2}}{\sigma_i \sigma_j}\right)$
- Different **sparsifications** :
  - 0 "Connected" graph : $w_{ij} = max(K(x_i, x_j), 0)$.
  - 1 k nearest-neighbor graph (with k=7).

⇒ Main questions :
  - Does basis expansion and RKHS help ?
  - Does "fusing" both x_i and Dx_i and work in a Sobolev space help ?
  - Does sparsification (that emphasizes the manifold hypothesis) help ?

Clustering procedures

- We test the different kernel/representation/sparsification using the two following clustering procedures. $\mathbf{S}$ is the Gram matrix.

Clustering procedures

- We test the different kernel/representation/sparsification using the two following clustering procedures. **S** is the Gram matrix.
  - **Kernel *k*-means (`K_km`)** :
    - Spectral decomposition of **S**.
    - Euclidean embedding : $\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_l \end{pmatrix}$ (all eigenvectors associated to strictly positive eigenvalues).
    - Apply *k*-means to **F**.

Clustering procedures

- We test the different kernel/representation/sparsification using the two following clustering procedures. **S** is the Gram matrix.
  - **Kernel k-means (`K_km`)** :
    - Spectral decomposition of **S**.
    - Euclidean embedding : $\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_l \end{pmatrix}$ (all eigenvectors associated to strictly positive eigenvalues).
    - Apply k-means to **F**.
  - **Spectral clustering (`SC_km`)** :
    - From **S**, determine **W** (with/without sparsification) and $\mathbf{L}_{sym}$.
    - Spectral decomposition of $\mathbf{L}_{sym}$.
    - Euclidean embedding : $\mathbf{F} = \begin{pmatrix} \mathbf{f}_1 & \dots & \mathbf{f}_k \end{pmatrix}$ (k first eigenvectors associated to the lowest eigenvalues).
    - Normalize rows of **F** to have unit norms.
    - Apply k-means to **F**.

Clustering procedures

- We test the different kernel/representation/sparsification using the two following clustering procedures. **S** is the Gram matrix.
  - **Kernel *k*-means (`K_km`)** :
    - Spectral decomposition of **S**.
    - Euclidean embedding : $\mathbf{F} = \begin{pmatrix}\mathbf{f}_1 & \dots & \mathbf{f}_l\end{pmatrix}$ (all eigenvectors associated to strictly positive eigenvalues).
    - Apply k-means to **F**.
  - **Spectral clustering (`SC_km`)** :
    - From **S**, determine **W** (with/without sparsification) and $\mathbf{L}_{sym}$.
    - Spectral decomposition of $\mathbf{L}_{sym}$.
    - Euclidean embedding : $\mathbf{F} = \begin{pmatrix}\mathbf{f}_1 & \dots & \mathbf{f}_k\end{pmatrix}$ (k first eigenvectors associated to the lowest eigenvalues).
    - Normalize rows of **F** to have unit norms.
    - Apply k-means to **F**.
- Baseline : kernel k-means with linear kernel $K_l(x_i, x_j) = \langle x_i, x_j \rangle_{\mathbb{L}^2}$.

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
`00_linear_K_km_`	$x_i \in \mathbb{L}^2([0, T])$	Linear	Ker. k-means	

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
00_linear_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
11_linear_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
01_linear_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Ker. k-means	

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
00_gaussian_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
11_gaussian_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
01_gaussian_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Ker. k-means	

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
00_linear_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
00_gaussian_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
11_linear_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
11_gaussian_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
01_linear_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Ker. k-means	
01_gaussian_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Ker. k-means	

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
00_linear_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
00_gaussian_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
11_linear_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
11_gaussian_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
01_linear_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Ker. k-means	
01_gaussian_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Ker. k-means	
00_linear_SC_km_0	$x_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	Connected
00_gaussian_SC_km_0	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	Connected
11_linear_SC_km_0	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	Connected
11_gaussian_SC_km_0	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	Connected
01_linear_SC_km_0	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Spectral Clust.	Connected
01_gaussian_SC_km_0	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Spectral Clust.	Connected

List of the 18 clustering models

Acronym	Representation	Kernel	Clustering proc.	Sparsif.
00_linear_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
00_gaussian_K_km_	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
11_linear_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Ker. k-means	
11_gaussian_K_km_	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Ker. k-means	
01_linear_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Ker. k-means	
01_gaussian_K_km_	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Ker. k-means	
00_linear_SC_km_0	$x_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	Connected
00_gaussian_SC_km_0	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	Connected
11_linear_SC_km_0	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	Connected
11_gaussian_SC_km_0	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	Connected
01_linear_SC_km_0	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Spectral Clust.	Connected
01_gaussian_SC_km_0	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Spectral Clust.	Connected
00_linear_SC_km_1	$x_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	7 near. neig.
00_gaussian_SC_km_1	$x_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	7 near. neig.
11_linear_SC_km_1	$Dx_i \in \mathbb{L}^2([0,T])$	Linear	Spectral Clust.	7 near. neig.
11_gaussian_SC_km_1	$Dx_i \in \mathbb{L}^2([0,T])$	Gaussian	Spectral Clust.	7 near. neig.
01_linear_SC_km_1	$x_i \in \mathbb{W}^{1,2}([0,T])$	Linear	Spectral Clust.	7 near. neig.
01_gaussian_SC_km_1	$x_i \in \mathbb{W}^{1,2}([0,T])$	Gaussian	Spectral Clust.	7 near. neig.

20 datasets from fda, fda.usc and UCR_TS_Archive_2015

Source	Type	Name	Nb of FD	Nb of Class	Nb of obs.
fda		Growth	93	2	31
fda.usc		poblenou	115	2	24
fda.usc		tecator	215	2	100
fda.usc		phoneme	250	5	150
UCR_TS	Spectro	Beef	30	5	470
UCR_TS	Simulated	CBF	30	3	128
UCR_TS	Spectro	Coffee	28	2	286
UCR_TS	ECG	ECG200	100	2	96
UCR_TS	Image	FaceFour	24	4	350
UCR_TS	Image	Fish	175	7	463
UCR_TS	Motion	GunPoint	50	2	150
UCR_TS	Sensor	Lightning2	60	2	637
UCR_TS	Sensor	Lightning7	70	7	319
UCR_TS	Image	MedicalImages	381	10	99
UCR_TS	Spectro	OliveOil	30	4	570
UCR_TS	Image	OSULeaf	200	6	427
UCR_TS	Image	SwedishLeaf	500	15	128
UCR_TS	Image	Symbols	25	6	398
UCR_TS	Sensor	Trace	100	4	275
UCR_TS	Simulated	TwoPatterns	1000	4	128

Clustering assessment and comparison

- Clustering models assessment :
  - **External validation** : for each dataset we have the ground truth.
  - Compare a clustering output against the ground truth using the **Normalized Mutual Information (NMI)** criterion. This measure is between 0 and 1 and the bigger the better.

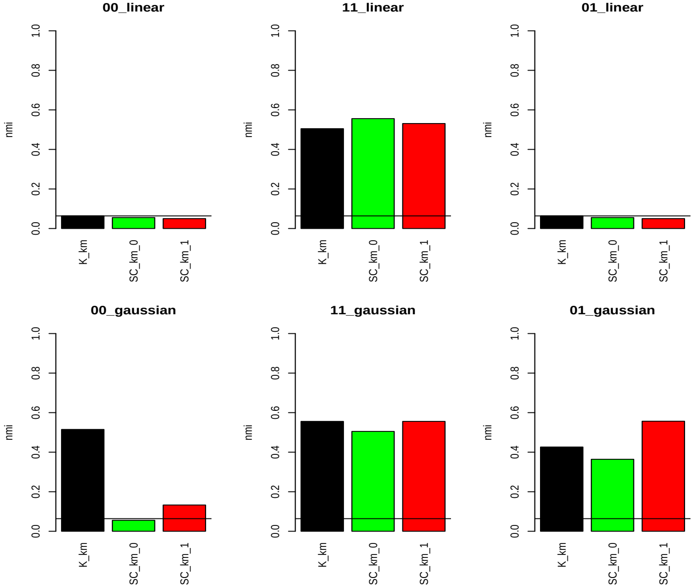
1. i beats j for a given dataset, if NMI of $i >$ NMI of j

Clustering assessment and comparison

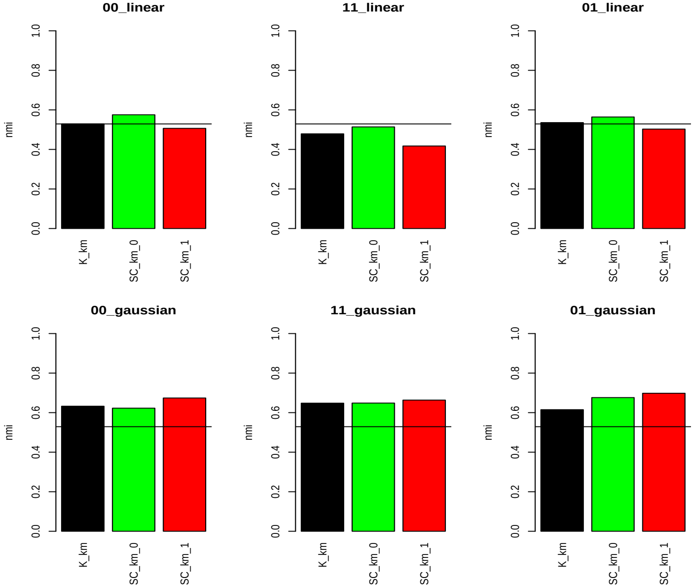
- Clustering models assessment :
  - **External validation** : for each dataset we have the ground truth.
  - Compare a clustering output against the ground truth using the **Normalized Mutual Information (NMI)** criterion. This measure is between 0 and 1 and the bigger the better.
- Comparing the 18 clustering models :
  - For each pair of clustering models (i, j), we count the **nb of times** i **beats**[1] j among the 20 datasets (each dataset is seen as a "match").
  - For an overall ranking of the clustering models, we use **Borda's voting rule : we rank according to the total nb of wins**. Each clustering model "plays" in total $20 \times 17 = 340$ "matches".

1. i beats j for a given dataset, if NMI of $i >$ NMI of j

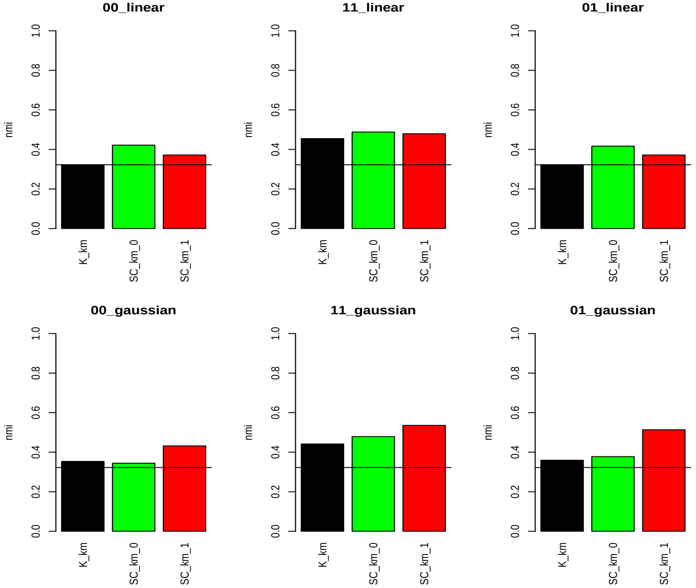
Examples of results : Growth data



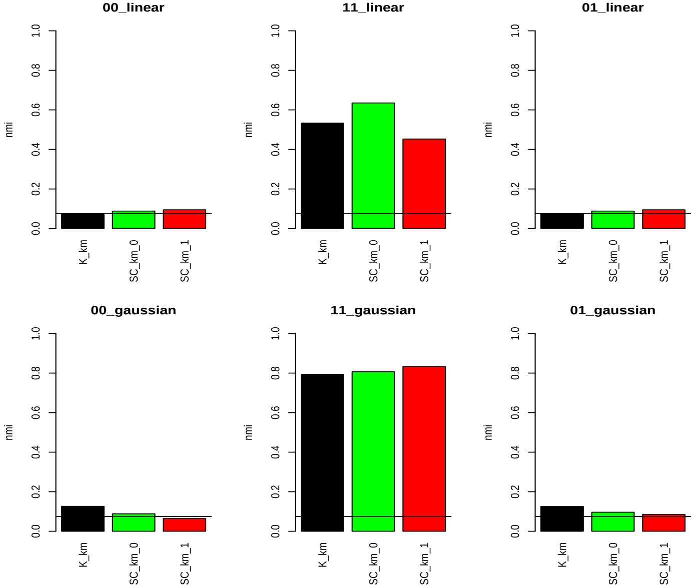
Examples of results : SwedishLeaf data



Examples of results : Fish data



Examples of results : Tecator data



Overall results : Borda's ranking

Rank	Clustering model	Nb of wins	Nb of losses
1	01_gaussian_SC_km_1	206	94
2	11_gaussian_K_km_	195	113
3	00_gaussian_K_km_	182	119
4	00_gaussian_SC_km_1	179	125
5	01_gaussian_SC_km_0	175	131
6	11_gaussian_SC_km_1	174	136
7	01_gaussian_K_km_	170	134
8	00_linear_SC_km_0	151	143
9	11_gaussian_SC_km_0	151	159
10	00_linear_SC_km_1	150	159
11	01_linear_SC_km_0	148	146
12	01_linear_SC_km_1	146	157
13	00_gaussian_SC_km_0	131	167
14	11_linear_SC_km_0	127	177
15	11_linear_K_km_	117	193
16	01_linear_K_km_	117	190
17	00_linear_K_km_	115	193
18	11_linear_SC_km_1	104	202

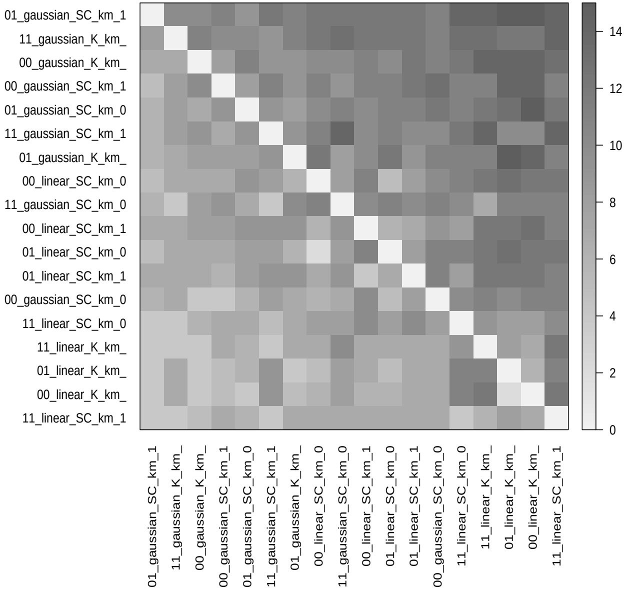
Overall results : Borda's ranking

Rank	Clustering model	Nb of wins	Nb of losses
1	01_gaussian_SC_km_1	206	94
2	11_gaussian_K_km_	195	113
3	00_gaussian_K_km_	182	119
4	00_gaussian_SC_km_1	179	125
5	01_gaussian_SC_km_0	175	131
6	11_gaussian_SC_km_1	174	136
7	01_gaussian_K_km_	170	134
9	11_gaussian_SC_km_0	151	159
13	00_gaussian_SC_km_0	131	167

Overall results : Borda's ranking

Rank	Clustering model	Nb of wins	Nb of losses
1	01_gaussian_SC_km_1	206	94
5	01_gaussian_SC_km_0	175	131
7	01_gaussian_K_km_	170	134

Borda's ranking visualization



Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.

Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.
  - Depending on the datasets x_i, Dx_i and (x_i, Dx_i) can give variable results, BUT (x_i, Dx_i) is never the worst performance of the three.

Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.
  - Depending on the datasets x_i, Dx_i and (x_i, Dx_i) can give variable results, BUT (x_i, Dx_i) is never the worst performance of the three.
  - With Gaussian kernel, $k = 7$ nearest neighbor graph sparsification always performs better than the "connected" graph sparsification. This outcome supports the manifold hypothesis.

Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.
  - Depending on the datasets x_i, Dx_i and (x_i, Dx_i) can give variable results, BUT (x_i, Dx_i) is never the worst performance of the three.
  - With Gaussian kernel, $k = 7$ nearest neighbor graph sparsification always performs better than the "connected" graph sparsification. This outcome supports the manifold hypothesis.
- Future work :

Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.
  - Depending on the datasets x_i, Dx_i and (x_i, Dx_i) can give variable results, BUT (x_i, Dx_i) is never the worst performance of the three.
  - With Gaussian kernel, $k = 7$ nearest neighbor graph sparsification always performs better than the “connected” graph sparsification. This outcome supports the manifold hypothesis.
- Future work :
  - x_i and Dx_i seems to bring complementary information BUT a “simple” fusion might degrade the overall performance.

Wrap up and future work

- Given a clustering procedure, say SC_km, we observe that :
  - Gaussian kernel gives better results than linear kernel.
  - Depending on the datasets x_i, Dx_i and (x_i, Dx_i) can give variable results, BUT (x_i, Dx_i) is never the worst performance of the three.
  - With Gaussian kernel, $k = 7$ nearest neighbor graph sparsification always performs better than the "connected" graph sparsification. This outcome supports the manifold hypothesis.
- Future work :
  - x_i and Dx_i seems to bring complementary information BUT a "simple" fusion might degrade the overall performance.

  $\Rightarrow$ Sparse clustering in Sobolev spaces : select discriminant features while performing the clustering.

Thank you for your attention !
Any question or comment ? :-)

Some references I

Abraham, C., Cornillon, P.-A., Matzner-Løber, E., and Molinari, N. (2003).
Unsupervised curve clustering using b-splines.
Scandinavian journal of statistics, 30(3) :581–595.

Bhatia, R. (2006).
Infinitely divisible matrices.
The American Mathematical Monthly, 113(3) :221–235.

Bouveyron, C. and Jacques, J. (2011).
Model-based clustering of time series in group-specific functional subspaces.
Advances in Data Analysis and Classification, 5(4) :281–300.

Chen, D., Müller, H.-G., et al. (2012).
Nonlinear manifold representations for functional data.
The Annals of Statistics, 40(1) :1–29.

Chung, F. R. (1997).
Spectral graph theory, volume 92.
American Mathematical Soc.

Dhillon, I. S., Guan, Y., and Kulis, B. (2004).
Kernel k-means : spectral clustering and normalized cuts.
In Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pages 551–556. ACM.

Ferraty, F. and Vieu, P. (2006).
Nonparametric functional data analysis : theory and practice.
Springer Science & Business Media.

Some references II

Fiedler, M. (1973).
Algebraic connectivity of graphs.
Czechoslovak mathematical journal, 23(2) :298–305.

Floriello, D. and Vitelli, V. (2017).
Sparse clustering of functional data.
Journal of Multivariate Analysis, 154 :1–18.

García, M. L. L., García-Ródenas, R., and Gómez, A. G. (2015).
K-means algorithms for functional data.
Neurocomputing, 151 :231–245.

Jacques, J. and Preda, C. (2014).
Functional data clustering : a survey.
Advances in Data Analysis and Classification, 8(3) :231–255.

Meila, M. and Shi, J. (2000).
Learning segmentation by random walks.
In NIPS, volume 14.

Muñoz, A. and González, J. (2010).
Representing functional data using support vector machines.
Pattern Recognition Letters, 31(6) :511–516.

Ng, A. Y., Jordan, M. I., Weiss, Y., et al. (2001).
On spectral clustering : Analysis and an algorithm.
In NIPS, volume 14, pages 849–856.

Some references III

Ramsay, J., Ramsay, J., and Silverman, B. (2005).
Functional Data Analysis.
Springer Science & Business Media.

Rossi, F. and Villa, N. (2006).
Support vector machine for functional data classification.
Neurocomputing, 69(7-9) :730–742.

Shi, J. and Malik, J. (2000).
Normalized cuts and image segmentation.
IEEE Transactions on pattern analysis and machine intelligence, 22(8) :888–905.

Verma, D. and Meila, M. (2003).
A comparison of spectral clustering algorithms.
University of Washington Tech Rep UWCSE030501, 1 :1–18.

Von Luxburg, U. (2007).
A tutorial on spectral clustering.
Statistics and computing, 17(4) :395–416.

Von Luxburg, U., Belkin, M., and Bousquet, O. (2008).
Consistency of spectral clustering.
The Annals of Statistics, pages 555–586.

Wang, J.-L., Chiou, J.-M., and Müller, H.-G. (2016).
Functional data analysis.
Annual Review of Statistics and Its Application, 3 :257–295.

Some references IV

Zelnik-Manor, L. and Perona, P. (2005).
Self-tuning spectral clustering.
In Advances in neural information processing systems, pages 1601–1608.