Regularized Multivariate Functional Principal Component Analysis

Mehdi Maadooliat
(Joint work with Yue Zhao and Dr. Hossein Haghbin)

Associate Professor
Department of Mathematical and Statistical Sciences
Marquette University
mehdi.maadooliat@mu.edu

5th Seminar on Spatial Statistics and Its Applications
IKIU, Qazvin, IR
(October 25, 2023)
Background: From FPCA to Regularized FPCA

- Performance of FPCA is often enhanced by regularization techniques.

**Figure:** Estimated first four principal components for the pinch force data.
Left: non-smoothed FPCs.
Right: smoothed FPCs
Motivating Example: Cursive Handwriting

fda – Cursive handwriting samples

non-noisy handwritings
Motivating Example: Cursive Handwriting

Cursive handwriting coordinates vs. time

X coordinate

Y coordinate

Mehdi Maadooliat

ReMFPCA

ReMFPCA
Motivating Example: Cursive Handwriting

Cursive handwriting coordinates vs. time

- Cursive handwritings
- MFPCA

X coordinate

Y coordinate

Time
Motivating Example: Cursive Handwriting

MFPCA – PC 1

- non-noisy handwritings
- non-noisy MFPCA

X coordinate

Y coordinate

Mehdi Maadooliat
Motivating Example: Cursive Handwriting

Cursive handwriting coordinates vs. time

- Cursive handwritings
- MFPCA

X coordinate

Y coordinate

time
Motivating Example: Cursive Handwriting

noisy handwriting coordinates vs. time

- noisy handwritings

X coordinate

Y coordinate

time
**Motivating Example: Cursive Handwriting**

noisy handwriting coordinates vs. time

<table>
<thead>
<tr>
<th>Cursive handwritings</th>
<th>MFPCA</th>
</tr>
</thead>
</table>

**X coordinate**

**Y coordinate**
Motivating Example: Cursive Handwriting

fda – noisy handwriting samples

noisy handwritings

X coordinate

Y coordinate

X coordinate
Motivating Example: Cursive Handwriting
Motivating Example: Cursive Handwriting

ReMFPCA – $\alpha = 9.536743e-07$

noisy handwritings
regularized MFPCA

X coordinate

Y coordinate
Motivating Example: Cursive Handwriting

ReMFPCA – $\alpha = 9.536743\times10^{-7}$

noisy handwritings
non-noisy MFPCA
regularized MFPCA

X coordinate
Y coordinate
Literature Review

- Rice and Silverman (1991) and Silverman (1996) are pioneer works in regularized FPCA (ReFPCA).
  - Studied functions in Hilbert space and developed roughness penalty based on derivative operators.
  - Mathematical foundation of the ReFPCA is developed in Sobolev spaces.

- Huang et al. (2008) proposed an alternative approach from the penalized SVD point of view.
  - Some nice computational properties.
  - A closed form of CV (GCV) criteria can be derived.

- Chiou et al. (2014) and Happ and Greven (2018) extend FPCA methods to multivariate version: Multivariate FPCA (MFPCA).
  - Enables exploring the relationships between multivariate functions.
Literature Review

- Rice and Silverman (1991) and Silverman (1996) are pioneer works in regularized FPCA (ReFPCA).
  - Studied functions in Hilbert space and developed roughness penalty based on derivative operators.
  - Mathematical foundation of the ReFPCA is developed in Sobolev spaces.

- Huang et al. (2008) proposed an alternative approach from the penalized SVD point of view.
  - Some nice computational properties.
  - A closed form of CV (GCV) criteria can be derived.

- Chiou et al. (2014) and Happ and Greven (2018) extend FPCA methods to multivariate version: Multivariate FPCA (MFPCA).
  - Enables exploring the relationships between multivariate functions.
Rice and Silverman (1991) and Silverman (1996) are pioneer works in regularized FPCA (ReFPCA).

- Studied functions in Hilbert space and developed roughness penalty based on derivative operators.
- Mathematical foundation of the ReFPCA is developed in Sobolev spaces.

Huang et al. (2008) proposed an alternative approach from the penalized SVD point of view.

- Some nice computational properties.
- A closed form of CV (GCV) criteria can be derived.

Chiou et al. (2014) and Happ and Greven (2018) extend FPCA methods to multivariate version: Multivariate FPCA (MFPCA).

- Enables exploring the relationships between multivariate functions.
Representation of the functional data (FD) in ReMFPCA

As Basis object:

@grid: vector \((x_1, x_2, \ldots, x_m)\)

@B: matrix 
\[
\begin{pmatrix}
    b_{11} & \cdots & b_{d1} \\
    \vdots & \ddots & \vdots \\
    b_{1m} & \cdots & b_{dm}
\end{pmatrix}
\]
Representation of the functional data (FD) in ReMFPCA

As fd object:

- **grid**: vector \((x_1, x_2, \ldots, x_m)\)
- **B**: matrix \[
\begin{pmatrix}
  b_{11} & \cdots & b_{d1} \\
  \vdots & \ddots & \vdots \\
  b_{1m} & \cdots & b_{dm}
\end{pmatrix}
\]
- **C**: matrix \[
\begin{pmatrix}
  \vdots \\
  \vdots \\
  \vdots \\
\end{pmatrix}
\]
Representation of the functional data (FD) in ReMFPCA

As fd object:

- \( \text{@grid: vector} \) \( (x_1, x_2, \ldots, x_m) \)
- \( \text{@B: matrix} \) \[
\begin{pmatrix}
 b_{11} & \cdots & b_{d1} \\
 \vdots & \ddots & \vdots \\
 b_{1m} & \cdots & b_{dm}
\end{pmatrix}
\]
- \( \text{@C: matrix} \) \[
\begin{pmatrix}
 c_{11} \\
 \vdots \\
 c_{1d}
\end{pmatrix}
\]
As ēd object:

- **@grid:** vector \( (x_1, x_2, \ldots, x_m) \)
- **@B:** matrix
  \[
  \begin{pmatrix}
  b_{11} & \cdots & b_{d1} \\
  \vdots & \ddots & \vdots \\
  b_{1m} & \cdots & b_{dm}
  \end{pmatrix}
  \]
- **@C:** matrix
  \[
  \begin{pmatrix}
  c_{11} & c_{21} \\
  \vdots & \vdots \\
  c_{1d} & c_{2d}
  \end{pmatrix}
  \]
Representation of the functional data (FD) in ReMFPCA

As fd object:

- **@grid**: vector \((x_1, x_2, \ldots, x_m)\)
- **@B**: matrix
  \[
  \begin{pmatrix}
  b_{11} & \cdots & b_{d1} \\
  \vdots & \ddots & \vdots \\
  b_{1m} & \cdots & b_{dm}
  \end{pmatrix}
  \]
- **@C**: matrix
  \[
  \begin{pmatrix}
  c_{11} & c_{21} & c_{31} & \cdots \\
  \vdots & \vdots & \vdots & \vdots \\
  c_{1d} & c_{2d} & c_{3d} & \cdots
  \end{pmatrix}
  \]
Representation of the functional data (FD) in ReMFPCA

### Basis object:

<table>
<thead>
<tr>
<th>@grid: matrix</th>
<th>[ \begin{pmatrix} x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x}, y_1, y_1, \ldots, y_1, y_2, y_2, \ldots, y_2, \ldots, y_{m_y} \end{pmatrix} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \hat{m} = m_x \times m_y )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>@B: matrix</th>
<th>[ \begin{pmatrix} b_{11} &amp; \cdots &amp; b_{dx,1} &amp; \cdots &amp; b_{d1} \ \vdots &amp; \ddots &amp; \vdots &amp; \ddots &amp; \vdots \ b_{1\hat{m}} &amp; \cdots &amp; b_{dx,\hat{m}} &amp; \cdots &amp; b_{d\hat{m}} \end{pmatrix} ]</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d = dx \times dy )</td>
<td></td>
</tr>
</tbody>
</table>
Representation of the functional data (FD) in ReMFPCA

As fd object:

- **@grid**: matrix

\[
\begin{pmatrix}
x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x} \\
y_1, y_1, \ldots, y_1, y_2, y_2, \ldots, y_2, \ldots, y_m
\end{pmatrix}
\]

- **@B**: matrix

\[
\begin{pmatrix}
b_{11} & \cdots & b_{dx,1} & \cdots & b_{d1} \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
b_{1m} & \cdots & b_{dx,m} & \cdots & b_{d1m}
\end{pmatrix}
\]

- **@C**: matrix

\[
\begin{pmatrix}
\end{pmatrix}
\]
Representation of the functional data (FD) in ReMFPCA

As fd object:

<table>
<thead>
<tr>
<th>@grid: matrix</th>
<th>( \begin{pmatrix} x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x} \ y_1, y_1, \ldots, y_1, y_2, y_2, \ldots, y_2, \ldots, y_{m_y} \end{pmatrix} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>@B: matrix</td>
<td>( \begin{pmatrix} b_{11} &amp; \ldots &amp; b_{d_1x,1} &amp; \ldots &amp; b_{d_1} \ \vdots &amp; \ddots &amp; \vdots &amp; \ddots &amp; \vdots \ b_{1m_1} &amp; \ldots &amp; b_{d_1,1m_1} &amp; \ldots &amp; b_{d_1m_1} \end{pmatrix} )</td>
</tr>
<tr>
<td>@C: matrix</td>
<td>( \begin{pmatrix} c_{11} \ \vdots \ c_{1d} \end{pmatrix} )</td>
</tr>
</tbody>
</table>
Representation of the functional data (FD) in ReMFPCA

As fd object:

<table>
<thead>
<tr>
<th>@grid: matrix</th>
<th>( \begin{pmatrix} x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x}, x_1, x_2, \ldots, x_{m_x} \ y_1, y_1, \ldots, y_{1}, y_{1}, y_2, y_2, \ldots, y_{2}, \ldots, y_{m_y} \end{pmatrix} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>@B: matrix</td>
<td>( \begin{pmatrix} b_{11} &amp; \cdots &amp; b_{dx,1} &amp; \cdots &amp; b_{d1} \ \vdots &amp; \vdots &amp; \vdots &amp; \vdots &amp; \vdots \ b_{1m} &amp; \cdots &amp; b_{dx,m} &amp; \cdots &amp; b_{dmm} \end{pmatrix} )</td>
</tr>
<tr>
<td>@C: matrix</td>
<td>( \begin{pmatrix} c_{11} &amp; c_{21} \ \vdots &amp; \vdots \ c_{1d} &amp; c_{2d} \end{pmatrix} )</td>
</tr>
</tbody>
</table>
Representation of the functional data (FD) in ReMFPCA

As fd object:

\[
\begin{align*}
@grid: & \quad \text{matrix} \\
(b_{11} & \ldots b_{dx,1} \ldots b_{d1}) \\
(c_{11} & \ldots c_{dx,1} \ldots c_{d1}) \\
(c_{1d} & \ldots c_{dx,d} \ldots c_{d,d})
\end{align*}
\]
How about Multivariate Functional Data (MFD) observed over different dimensional domains?

As fd object:

\[
\begin{align*}
@C: & \text{list} \\
@\text{grid}: & \{(x_1, x_2, \ldots, x_m) \\
& (y_{11}, y_{12}, \ldots, y_{1m}, y_{21}, y_{22}, \ldots, y_{2m}) \ldots \}
\end{align*}
\]

\[
\begin{align*}
@B: & \text{list} \\
& (b_{11}, \ldots, b_{d1}) \\
& \vdots \\
& (b_{lm}, \ldots, b_{dm}) \\
& (b_{l1}, \ldots, b_{dx1}, \ldots, b_{d1}) \\
& \vdots \\
& (b_{lm}, \ldots, b_{dxm}, \ldots, b_{dm})
\end{align*}
\]

\[
\begin{align*}
@C: & \text{list} \\
& \left( \begin{array}{c}
\vdots \\
\vdots \\
\vdots
\end{array} \right)
\end{align*}
\]
How about Multivariate Functional Data (MFD) observed over different dimensional domains?

As \( \text{fd} \) object:

- \( \mathbf{C} : \) list
- \( \mathbf{grid} : \) \( \mathbf{x} \), \( x_1, x_2, \ldots, x_m \)
- \( \mathbf{B} : \) list
  - \( b_{11}, \ldots, b_{d1} \)
  - \( \vdots \)
  - \( b_{1m}, \ldots, b_{dm} \)
- \( \mathbf{C} : \) list
  - \( c_{11}, \ldots, c_{1d} \)
  - \( \vdots \)
  - \( c_{1d} \)
Representation of the functional data (FD) in ReMFPCA

- How about Multivariate Functional Data (MFD) observed over different dimensional domains?

As fd object:

\[ \text{grid: list} \]
\[ \text{B: list} \]
\[ \text{C: list} \]

\[ \begin{pmatrix} b_{11} & \ldots & b_{d1} \\ \vdots & \ddots & \vdots \\ b_{1m} & \ldots & b_{dm} \end{pmatrix} \]

\[ \begin{pmatrix} c_{11} & c_{21} \\ \vdots & \vdots \\ c_{1d} & c_{2d} \end{pmatrix} \]
How about Multivariate Functional Data (MFD) observed over different dimensional domains?

As fd object:

\[ \text{@C: list } \]
\[ \text{@grid: } \begin{array}{c} x_1, x_2, \ldots, x_m \\ (x_1, x_2, \ldots, x_m, x_1, x_2, \ldots, x_m, x_1, x_2, \ldots, x_m) \\ (y_1, y_2, \ldots) \end{array} \]
\[ \text{@B: list } \]
\[ \begin{pmatrix} b_{11} & \cdots & b_{d1} \\ \vdots & \ddots & \vdots \\ b_{1m} & \cdots & b_{dm} \end{pmatrix} \]
\[ \text{@C: list } \]
\[ \begin{pmatrix} c_{11} & c_{21} & c_{31} & \cdots \\ \vdots & \vdots & \vdots & \vdots \\ c_{1d} & c_{2d} & c_{3d} & \cdots \end{pmatrix} \]
Regularized MFPCA

- Regularized MFPCA (ReMFPCA) seems an intuitive next step to enhance the performance of MFPCA.

- Two ReMFPCA approaches are proposed by our research group.
  - **Regularized Eigen Decomposition of the Covariance Operator:**
    By extending Silverman (1996) approach into a multivariate framework.
    (Submitted: https://doi.org/10.48550/arXiv.2306.13980)

  - **Penalized Functional SVD (fSVD) of the Data Operator:**
    We study theoretical foundations and implementation of fSVD for MFPCA. Specifically we extend Huang et al. (2008) approach to the multivariate setup in Sobolev space with
    - Flexibility in tuning parameters selection, and
    - Computation efficiency.

    (Ongoing Project: focus of today's talk)
Regularized MFPCA

- Regularized MFPCA (ReMFPCA) seems an intuitive next step to enhance the performance of MFPCA.

- Two ReMFPCA approaches are proposed by our research group.
  - **Regularized Eigen Decomposition of the Covariance Operator:**
    By extending Silverman (1996) approach into a multivariate framework. (Submitted: https://doi.org/10.48550/arXiv.2306.13980)
  
  - **Penalized Functional SVD (fSVD) of the Data Operator:**
    We study theoretical foundations and implementation of fSVD for MFPCA. Specifically we extend Huang et al. (2008) approach to the multivariate setup in Sobolev space with
    - Flexibility in tuning parameters selection, and
    - Computation efficiency.

  (Ongoing Project: focus of today's talk)
Regularized MFPCA

- Regularized MFPCA (ReMFPCA) seems an intuitive next step to enhance the performance of MFPCA.

- Two ReMFPCA approaches are proposed by our research group.
  - **Regularized Eigen Decomposition of the Covariance Operator:**
    By extending Silverman (1996) approach into a multivariate framework. (Submitted: https://doi.org/10.48550/arXiv.2306.13980)
  - **Penalized Functional SVD (fSVD) of the Data Operator:**
    We study theoretical foundations and implementation of fSVD for MFPCA. Specifically we extend Huang et al. (2008) approach to the multivariate setup in Sobolev space with
    - Flexibility in tuning parameters selection, and
    - Computation efficiency.

(Ongoing Project: focus of today’s talk)
Preliminary Notations

- Let $H_j$ to be a Hilbert space equipped with the inner product
  \[ \langle x, y \rangle_{H_j} = \int_{\mathcal{T}_j} x(t)y(t)dt, \quad \text{where} \ x, y \in H_j \ \text{and} \ j = 1, \cdots, p. \]

- The Sobolev space $W^2_j$ is defined as
  \[ W^2_j := \{ x(\cdot) : x \ \text{and} \ x' \ \text{are absolutely continuous on} \ \mathcal{T}_j \ \text{and} \ x'' \in H_j \}. \]

- Given a smoothing parameter $\alpha_j > 0$, we can define the inner product
  \[ \langle x, y \rangle_{\alpha_j} := \langle x, y \rangle_{H_j} + \alpha_j \langle x'', y'' \rangle_{H_j}. \]
  The $\alpha_j$-orthogonality in Sobolev space $W^2_j$ is
  \[ \langle x_j, y_j \rangle_{\alpha_j} = 0. \]
Preliminary Notations

• Let $H_j$ to be a Hilbert space equipped with the inner product

$$\langle x, y \rangle_{H_j} = \int_{T_j} x(t)y(t)dt,$$

where $x, y \in H_j$ and $j = 1, \cdots , p$.

• The Sobolev space $W^2_j$ is defined as

$$W^2_j := \{x(\cdot) : x and x' are absolutely continuous on T_j and x'' \in H_j\}.$$

• Given a smoothing parameter $\alpha_j > 0$, we can define the inner product

$$\langle x, y \rangle_{\alpha_j} := \langle x, y \rangle_{H_j} + \alpha_j \langle x'', y'' \rangle_{H_j}.$$

The $\alpha_j$-orthogonality in Sobolev space $W^2_j$ is

$$\langle x_j, y_j \rangle_{\alpha_j} = 0.$$
Preliminary Notations cont.

- Define the cartesian Hilbert product space $\mathbb{H} := H_1 \times \cdots \times H_p$, where each $H_j$ is a Hilbert space.
  For $\mathbf{x} = (x_1, \cdots, x_p)$ and $\mathbf{y} = (y_1, \cdots, y_p) \in \mathbb{H}$,
  \[
  \langle \mathbf{x}, \mathbf{y} \rangle_{\mathbb{H}} = \sum_{j=1}^{p} \langle x_j, y_j \rangle_{H_j}.
  \]

- Define the cartesian Sobolev product spaces $\mathbb{W}^2 := W_1^2 \times \cdots \times W_p^2$.
  For $\mathbf{x}, \mathbf{y} \in \mathbb{W}^2$ and smoothing parameter $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_p) \in \mathbb{R}^p$,
  \[
  \langle \mathbf{x}, \mathbf{y} \rangle_\alpha = \sum_{j=1}^{p} \langle x_j, y_j \rangle_{\alpha_j}.
  \]
  The $\alpha$-orthogonality in Sobolev space $\mathbb{W}^2$ is $\langle \mathbf{x}, \mathbf{y} \rangle_\alpha = 0$. 

Preliminary Notations cont.

- Define the cartesian Hilbert product space $\mathbb{H} := H_1 \times \cdots \times H_p$, where each $H_j$ is a Hilbert space. For $x = (x_1, \cdots, x_p)$ and $y = (y_1, \cdots, y_p) \in \mathbb{H}$,

$$\langle x, y \rangle_{\mathbb{H}} = \sum_{j=1}^{p} \langle x_j, y_j \rangle_{H_j}.$$

- Define the cartesian Sobolev product spaces $\mathbb{W}^2 := W_1^2 \times \cdots \times W_p^2$. For $x, y \in \mathbb{W}^2$ and smoothing parameter $\alpha = (\alpha_1, \alpha_2, \cdots, \alpha_p) \in \mathbb{R}^p$,

$$\langle x, y \rangle_{\alpha} = \sum_{j=1}^{p} \langle x_j, y_j \rangle_{\alpha_j}.$$

The $\alpha$-orthogonality in Sobolev space $\mathbb{W}^2$ is $\langle x, y \rangle_{\alpha} = 0$. 
Theorem

Denote $x_i := [x_{i,j}]_{j=1}^p \in \mathbb{H}$ and the data operator $X := [x_i]_{i=1}^n \in \mathbb{F}^{p \times n}$ with rank $m \leq n$. There exist linearly independent elements $\phi_1, \cdots, \phi_m$ from $\mathbb{H}$ and $v_1, \cdots, v_m$ from $\mathbb{R}^n$ that are orthonormal and

$$X = \sum_{i=1}^m \sqrt{\lambda_i} \, v_i \otimes \phi_i,$$

where $\lambda_i$’s are non-ascending positive scalars.

The goal is to obtain regularized FPCs, which is equivalent to solve the following penalized functional SVD problem:

$$\min_{\phi: \|\phi\|_{\alpha} = 1, \, v \in \mathbb{R}^n} \|X - v \otimes \phi\|_F^2 + v^T v \sum_{j=1}^p \alpha_j \langle \phi_j'', \phi_j'' \rangle_{H_j}$$

(2)
Functional SVD and penalized functional SVD

Theorem

Denote \( x_i := [x_{i,j}]_{j=1}^p \in \mathbb{H} \) and the data operator \( X := [x_i]_{i=1}^n \in \mathbb{F}^{p \times n} \) with rank \( m \leq n \). There exist linearly independent elements \( \varphi_1, \cdots, \varphi_m \) from \( \mathbb{H} \) and \( v_1, \cdots, v_m \) from \( \mathbb{R}^n \) that are orthonormal and

\[
X = \sum_{i=1}^{m} \sqrt{\lambda_i} \ v_i \otimes \varphi_i, \quad (1)
\]

where \( \lambda_i \)'s are non-ascending positive scalars.

The goal is to obtain regularized FPCs, which is equivalent to solve the following penalized functional SVD problem:

\[
\min_{\varphi: \|\varphi\|_\alpha = 1, \ v \in \mathbb{R}^n} \|X - v \otimes \varphi\|_F^2 + v^T \sum_{j=1}^{p} \alpha_j \langle \varphi''_j, \varphi''_j \rangle_{H_j} \quad (2)
\]
Finite dimensional representation of functional data

- In implementation, each functional observations are considered as projection on a finite dimensional subspace \( H^d_{j} = sp\{v^k_j\}_{k=1}^{d_j} \subseteq H_j \). And we define \( \mathbb{H}^d := H^d_1 \times \cdots \times H^d_p \).

- The minimization problem given in (2) becomes

\[
\min_{\sim b, \sim v} \| \sim B - \sim v \sim b^\top \|_F^2 + \sim v^\top \sim v \sim b^\top \sim \Omega \sim \alpha \sim \sim b, \quad (3)
\]

where \( \sim B = B \sim G^{\frac{1}{2}}, \sim b = \sim G^{\frac{1}{2}} b, \sim \Omega \sim \alpha = \sim G^{-\frac{1}{2}} \sim D \sim \alpha \sim G^{-\frac{1}{2}}, \) and

- \( B \) is the matrix associated to the projection coefficients of \( \chi \) on \( \mathbb{H}^d \),

- \( G := \text{diag} \{G_1, \cdots, G_p\} \), where

\[
G_j = [\langle v^l_j, v^k_j \rangle_{H_j}]_{l,k=1}^{d_j},
\]

- \( b \) is the vector corresponding to the projection coefficients of \( \varphi \) on \( \mathbb{H}^d \),

- \( D \sim \alpha := \text{diag}\{\alpha_1 \sim D_1, \cdots, \alpha_p \sim D_p\} \), where

\[
D_j = [\langle v^{l''}_j, v^{k''}_j \rangle_{H_j}]_{l,k=1}^{d_j}.
\]
Finite dimensional representation of functional data

- In implementation, each functional observations are considered as projection on a finite dimensional subspace $H^d_j = sp\{v^k_j\}_{k=1}^{d_j} \subseteq H_j$.
  And we define $\mathbb{H}^d := H_1^d \times \cdots \times H_p^d$.
- The minimization problem given in (2) becomes
  
  $\min_{b,\nu} \|B - \nu b^\top\|_F^2 + \nu^\top \nu b^\top \Omega_\alpha b$,  
  
  where $B \sim B G^\frac{1}{2}$, $b \sim G^\frac{1}{2} b$, $\Omega_\alpha \sim G^{-\frac{1}{2}} D_\alpha G^{-\frac{1}{2}}$, and
- $B$ is the matrix associated to the projection coefficients of $X$ on $\mathbb{H}^d$,
- $G := \text{diag} \{G_1, \cdots, G_p\}$, where 
  
  $G_j = [\langle v^l_j, v^k_j \rangle_{H_j}]_{l,k=1}^{d_j}$,
- $b$ is the vector corresponding to the projection coefficients of $\varphi$ on $\mathbb{H}^d$,
- $D_\alpha := \text{diag}\{\alpha_1 D_1, \cdots, \alpha_p D_p\}$, 
  where $D_j = [\langle v^l_j'', v^k_j'' \rangle_{H_j}]_{l,k=1}^{d_j}$. 

Mehdi Maadooliat

ReMFPCA

ReMFPCA
Implementation Strategy: Power algorithm

- To optimize (3), one may use the following iterative power algorithm:
  1. Initialize $b$.
  2. Repeat until convergence:
     3. $v \leftarrow B G b$,
     4. $b \leftarrow S_\alpha^2 G B^T v$,
     5. normalize $b$.

Here $S_\alpha = (G + D_\alpha)^{-\frac{1}{2}}$ is referred to a half-smoothing matrix.

- For a fixed $v$, the penalized SVD in (3) becomes a penalized regression problem:

$$\|\tilde{y} - \tilde{X} b\|^2 + b^T (v^T v \Omega_\alpha) b,$$

where

$$\tilde{y} := \left[ B_{\sim,1}^T, B_{\sim,2}^T, \ldots, B_{\sim,d}^T \right]^T \in \mathbb{R}^{nd}, \quad \tilde{X} := \begin{bmatrix} v \\ \vdots \\ v \end{bmatrix} \in \mathbb{R}^{nd \times d}.$$
Implementation Strategy: Power algorithm

To optimize (3), one may use the following iterative power algorithm:

1. Initialize $b$.
2. Repeat until convergence:
   a. $v \leftarrow B G b$,
   b. $b \leftarrow S_{\alpha}^2 G b^T v$,
   c. normalize $b$.

Here $S_{\alpha} = (G + D_{\alpha})^{-\frac{1}{2}}$ is referred to a half-smoothing matrix.

For a fixed $v$, the penalized SVD in (3) becomes a penalized regression problem:

$$\|\bar{y} - \bar{X} b\|^2 + b^T (v^T v \Omega_{\alpha}) b,$$

where

$$\bar{y} := \left[ B_{\sim 1}^T, B_{\sim 2}^T, \ldots, B_{\sim d}^T \right]^T \in \mathbb{R}^{nd}, \quad \bar{X} := \begin{bmatrix} v \\ \vdots \\ v \end{bmatrix} \in \mathbb{R}^{nd \times d}.$$
Implementation Strategy: Power algorithm

To optimize (3), one may use the following iterative power algorithm:

1. Initialize $b$.
2. Repeat until convergence:
   a. $v \leftarrow B G b$,
   b. $b \leftarrow S_\alpha^2 G B^\top v$,
   c. normalize $b$.

Here $S_\alpha = (G + D_\alpha)^{-\frac{1}{2}}$ is referred to a half-smoothing matrix.

For a fixed $v$, the penalized SVD in (3) becomes a penalized regression problem:

$$\|\tilde{y} - \bar{X} b\|^2 + b^\top (v^\top v \Omega_\alpha) b,$$

where

$$\tilde{y} := \begin{bmatrix} B_{1.}^\top, B_{2.}^\top, \ldots, B_{d.}^\top \end{bmatrix}^\top \in \mathbb{R}^{nd}, \quad \bar{X} := \begin{bmatrix} v \\ \vdots \\ v \end{bmatrix} \in \mathbb{R}^{nd \times d}. $$
Tuning parameters selection based on GCV

The GCV criteria can be simply nested within the power algorithm

\[
GCV_\alpha = \frac{1}{d} \sum_{k=1}^{p} \frac{\| (I_k - \tilde{S}_{\alpha_k})(\tilde{B}_k^T v) \|^2}{(1 - \frac{1}{d} \text{tr}\{\tilde{S}_{\alpha_k}\})^2},
\]

where \( \tilde{S}_{\alpha_k} \) is \( k^{th} \) diagonal block of \( \tilde{S}_\alpha := G^2 S_\alpha^2 G^2 \)

\(a)\ v \leftarrow B G b \)  Simply nest GCV selection of \( \alpha \) inside step (b)

\(b)\ b \leftarrow S_\alpha^2 G B^T v \)

\(c)\ Normalize \ b \)
Two flexible choices in power algorithm

- **Simultaneous power algorithm:**
  - Obtaining FPCs jointly where all FPCs share the same tuning parameter.
  - Preserves the $\alpha$-orthogonality in Sobolev space.
  - Since we compute the $(p > 1)$-dimensional subspace simultaneously, a QR factorization is needed in step 2(b).

- **Sequential power algorithm:**
  - Obtaining FPCs sequentially where different tuning parameter is allowed for each FPC.
  - The flexibility of having different level of smoothness for FPCs.
  - Losing the $\alpha$-orthogonality property in Sobolev space.
Two flexible choices in power algorithm

- **Simultaneous power algorithm:**
  - Obtaining FPCs jointly where all FPCs share the same tuning parameter.
  - Preserves the $\alpha$-orthogonality in Sobolev space.
  - Since we compute the $(p > 1)$-dimensional subspace simultaneously, a QR factorization is needed in step 2(b).

- **Sequential power algorithm:**
  - Obtaining FPCs sequentially where different tuning parameter is allowed for each FPC.
  - The flexibility of having different level of smoothness for FPCs.
  - Losing the $\alpha$-orthogonality property in Sobolev space.
Simulation setup

Let \( X(t) \) be a bivariate functional observation. We define a bivariate orthonormal basis system \( \psi_m(t) \), where

\[
\psi_m^{(1)}(t) = \sin ((2m - 1)\pi t) \quad \text{and} \quad \psi_m^{(2)}(t) = \sin \left( \frac{(4m - 3)\pi}{2} t \right).
\]

We adopt the following functional data generating model:

\[
X_i(t) = \sum_{m=1}^{M} \rho_{i,m} \psi_m(t), \quad \rho_{i,m} \sim \mathcal{N}(0, \lambda_m), \quad i = 1, \ldots, n. \tag{5}
\]

The goal is to examine scenarios where varying levels of noise are added to each \( \psi_m(t) \), where \( \tilde{\psi}_m(t) = \psi_m(t) + \epsilon_m(t) \).

We simulate our observations, using

\[
Y_i(t) = \sum_{m=1}^{M} \rho_{i,m}(\tilde{\psi}_m(t)), \tag{6}
\]
Simulation setup

- Let $\mathbf{X}(t)$ be a bivariate functional observation. We define a bivariate orthonormal basis system $\psi_m(t)$, where

$$
\psi_m^{(1)}(t) = \sin((2m - 1)\pi t) \quad \text{and} \quad \psi_m^{(2)}(t) = \sin\left(\frac{(4m - 3)\pi}{2} t\right).
$$

We adopt the following functional data generating model:

$$
\mathbf{X}_i(t) = \sum_{m=1}^{M} \rho_{i,m} \psi_m(t), \quad \rho_{i,m} \sim \mathcal{N}(0, \lambda_m), \quad i = 1, \ldots, n. \quad (5)
$$

- The goal is to examine scenarios where varying levels of noise are added to each $\psi_m(t)$, where $\tilde{\psi}_m(t) = \psi_m(t) + \epsilon_m(t)$.

- We simulate our observations, using

$$
\mathbf{Y}_i(t) = \sum_{m=1}^{M} \rho_{i,m}(\tilde{\psi}_m(t)), \quad (6)
$$
Simulation setup

- Let $X(t)$ be a bivariate functional observation. We define a bivariate orthonormal basis system $\psi_m(t)$, where

$$
\psi_m^{(1)}(t) = \sin ((2m - 1)\pi t)
$$

and

$$
\psi_m^{(2)}(t) = \sin \left( \frac{(4m - 3)\pi}{2} t \right).
$$

We adopt the following functional data generating model:

$$
X_i(t) = \sum_{m=1}^{M} \rho_{i,m} \psi_m(t), \quad \rho_{i,m} \sim \mathcal{N}(0, \lambda_m), \quad i = 1, \ldots, n. \quad (5)
$$

- The goal is to examine scenarios where varying levels of noise are added to each $\psi_m(t)$, where $\tilde{\psi}_m(t) = \psi_m(t) + \epsilon_m(t)$.

- We simulate our observations, using

$$
Y_i(t) = \sum_{m=1}^{M} \rho_{i,m} (\tilde{\psi}_m(t)), \quad (6)
$$
Simulation: Comparison

- To assess the performance of our sequential and joint approach, we compare them with two other methods: non-regularized MFPCA and Happ’s approach (Happ and Greven, 2018).

- The accuracy of the estimated eigenvalue and eigenfunction pairs, denoted as \( \hat{\lambda}_m \) and \( \hat{\psi}_m \) respectively, was evaluated by comparing them to their original counterparts:

  \[
  Err(\hat{\lambda}_m) = \frac{|\hat{\lambda}_m - \lambda_m|}{|\lambda_m|} \quad \text{and} \quad Err(\hat{\psi}_m) = \|\hat{\psi}_m - \psi\|_H.
  \]

- Furthermore, the accuracy of the estimates for each replication is assessed using the mean relative absolute error (MRAE), defined as

  \[
  \text{MRAE} = \frac{1}{n} \sum_{i=1}^{n} \frac{\|\hat{x}_i - x_i\|_H}{\|x_i\|_H},
  \]

  where \( \hat{x}_i = \sum_{m=1}^{J} \langle y_i, \hat{\psi}_m \rangle_H \hat{\psi}_m \).
Simulation: Comparison

- To assess the performance of our sequential and joint approach, we compare them with two other methods: non-regularized MFPCA and Happ’s approach (Happ and Greven, 2018).

- The accuracy of the estimated eigenvalue and eigenfunction pairs, denoted as $\hat{\lambda}_m$ and $\hat{\psi}_m$ respectively, was evaluated by comparing them to their original counterparts:

$$Err(\hat{\lambda}_m) = |\hat{\lambda}_m - \lambda_m|/|\lambda_m| \quad \text{and} \quad Err(\hat{\psi}_m) = ||\hat{\psi}_m - \psi||_H.$$  

- Furthermore, the accuracy of the estimates for each replication is assessed using the mean relative absolute error (MRAE), defined as

$$\text{MRAE} = \frac{1}{n} \sum_{i=1}^{n} (||\hat{x}_i - x_i||_H)/||x_i||_H,$$

where $\hat{x}_i = \sum_{m=1}^{J} \langle y_i, \hat{\psi}_m \rangle_H \hat{\psi}_m$. 
Simulation: Comparison

- To assess the performance of our sequential and joint approach, we compare them with two other methods: non-regularized MFPCA and Happ’s approach (Happ and Greven, 2018).

- The accuracy of the estimated eigenvalue and eigenfunction pairs, denoted as $\hat{\lambda}_m$ and $\hat{\psi}_m$ respectively, was evaluated by comparing them to their original counterparts:

  $$Err(\hat{\lambda}_m) = \frac{|\hat{\lambda}_m - \lambda_m|}{|\lambda_m|} \quad \text{and} \quad Err(\hat{\psi}_m) = ||\hat{\psi}_m - \psi||_H.$$  

- Furthermore, the accuracy of the estimates for each replication is assessed using the mean relative absolute error (MRAE), defined as

  $$\text{MRAE} = \frac{1}{n} \sum_{i=1}^{n} \frac{(||\hat{x}_i - x_i||_H)/||x_i||_H}{||x_i||_H},$$  

  where $\hat{x}_i = \sum_{m=1}^{J} \langle y_i, \hat{\psi}_m \rangle H \hat{\psi}_m$. 


Comparison result (for different trend patterns in eigenvalues)

Uniform levels of PC roughness

Figure: $\text{Err}(\hat{\psi}_m)$
Comparison result (for different trend patterns in eigenvalues)

**Figure:** $Err(\hat{\psi}_m)$

### Disparate levels of PC roughness

- **PC 1**
  - Linear
  - Decreasing
  - Exponential
  - Decreasing
  - Linear
  - Increasing
  - Exponential

- **PC 2**
  - Linear
  - Decreasing
  - Exponential
  - Decreasing
  - Linear
  - Increasing
  - Exponential

- **PC 3**
  - Linear
  - Decreasing
  - Exponential
  - Decreasing
  - Linear
  - Increasing
  - Exponential

- **PC 4**
  - Linear
  - Decreasing
  - Exponential
  - Decreasing
  - Linear
  - Increasing
  - Exponential

**Legend:**
- MFPCA
- Joint
- Sequential
- Happ
Comparison result (for different trend patterns in eigenvalues)

Figure: $Err(\hat{\lambda}_m)$
Comparison result
(for different trend patterns in eigenvalues)

Disparate levels of PC roughness

Figure: $Err(\hat{\lambda}_m)$
Comparison result (for different trend patterns in eigenvalues)

**Uniform levels of PC roughness**

- Linear decreasing
- Exponential decreasing
- Linear increasing
- Exponential increasing

**Figure: MRAE**
Comparison result (for different trend patterns in eigenvalues)

Disparate levels of PC roughness

Figure: MRAE
Consider a bivariate functional data that include active power and voltage consumption of one household in Sceaux (7km of Paris, France) between December 2006 and November 2010.
Interpretation of PC scores: FPC1

Left top: Average temperature heatmap; Left bottom: Clustering based on PC1 scores; Right: Clustering details on original data.
Interpretation of PC scores: FPC3

Figure: Boxplot of PC3 scores

Figure: Clustering details on original data.
Conclusion

- We developed ReMFPCA based on regularized functional SVD approach.

- An efficient power algorithm is proposed with two flexible choices:
  - Simultaneous power method: Jointly estimates all FPCs with a common smoothing parameter (FPCs will have the $\alpha$-orthogonality in Sobolev space).
  - Sequential power method: Estimating each FPC sequentially, where different smoothing parameters are allowed for each FPC (we will lose the $\alpha$-orthogonal property).

- A closed form GCV is derived from the regularized functional SVD approach, where it can significantly improve computational efficiency.
  - Proposed GCV criteria can be embedded within the power algorithm.
Thank you!

- Collaborators
  - Yue Zhao, Ph.D. Candidate, Marquette University
  - Hossein Haghbin, Assistant Professor, Persian Gulf University

Questions?
References


