A New Computationally Efficient Algorithm to solve Feature Selection for Functional Data Classification in High-dimensional Spaces

Tobia Boschi, Francesca Bonin, Rodrigo Ordonez-Hurtado,

Alessandra Pascale, Jonathan Epperlein

IBM Research, Europe

ICML Wien, 2024

A New Computationally Efficient Algorithm to solve Feature Selection for Functional Data Classification in High-dimensional Spaces

Tobia Boschi, Francesca Bonin, Rodrigo Ordonez-Hurtado,

Alessandra Pascale, Jonathan Epperlein

github.com/IBM/funGCN

IBM Research

Motivation: SEURO Project

Evaluate the **impact** of a **Digital Health solution** for managing multiple diseases in people over 60 across Europe.

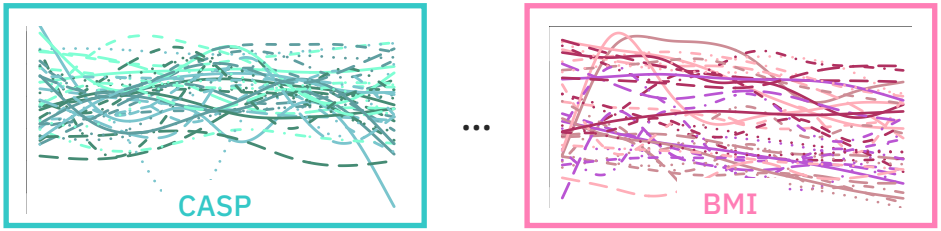
Challenges

Heterogeneous variables, Small sample size,

Complex longitudinal variables

Multiple variables for multiple patients

collected at different frequencies and timings



Goal: FSFC Feature Selection for Functional Classification

Develop a new highly efficient algorithm to solve feature selection in instances characterized by multivariate longitudinal variable and binary categorical responses

Motivation: SEURO Project

Evaluate the **impact** of a **Digital Health solution** for managing multiple diseases in people over 60 across Europe.

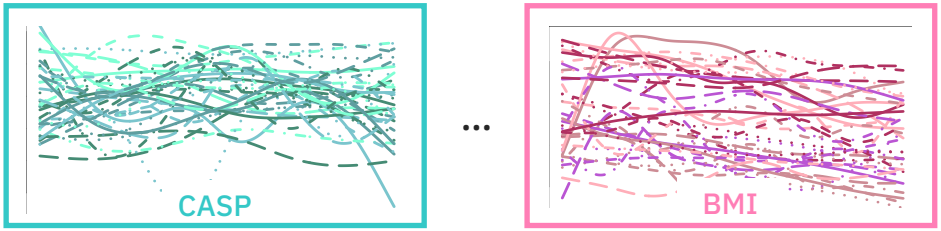
Challenges

Heterogeneous variables, Small sample size,

Complex longitudinal variables

Multiple variables for multiple patients

collected at different frequencies and timings



Strategy

(1) Represents the functions in a finite space: **Matrix Representation**

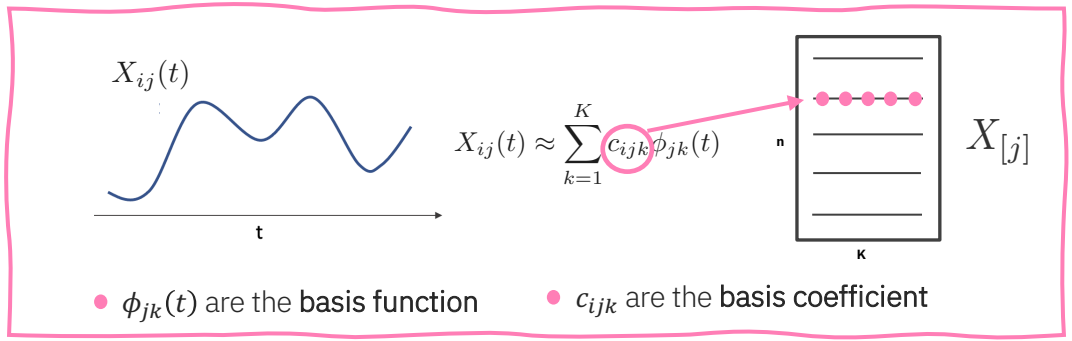
(2) Develop a **new optimization algorithm** in the multivariate framework, incorporating **logistic loss** and **elastic net penalty** to create sparsity

Matrix representation

Aim: represent each **functional variable** as a **matrix** of dimension **n** (number of samples) x **K**

$$\left[X_{[1]}\middle|\ldots\middle|X_{[p]}\right]$$

Given the functional variable $X_j(t) = (X_{1j}(t), \ldots, X_{nj}(t))$, we express each curve with a **basis expansion:**



(●) $\boldsymbol{\phi(t)}$: **FPC Functional Principal Components**

- **orthonormal:** improve *computational efficiency*
- **parsimonious:** *small K* (< 10) allows to reconstruct the curves

Optimization problem

adaptive weights LASSO Ridge

$$\min_{B} \left[\sum_{i=1}^{n} \log\left(1 + \exp\left(-Y_i \cdot \left(X_{(i)} B\right)\right)\right) + \sum_{j=1}^{p} \omega_j \left(\lambda_1 \|B_j\|_2 + \frac{\lambda_2}{2} \|B_j\|_2^2 \right) \right]$$

$h(XB)$: logistic loss

$\pi(B)$: elastic net penalty

Dual Augmented Lagrangian

$$\mathcal{L}_\sigma(V, Z, B) = h^*(V) + \pi^*(Z) - \sum_{j=1}^{p} \langle B_j, V^T X_j + Z_j \rangle + \frac{\sigma}{2} \sum_{j=1}^{p} \|V^T X_j + Z_j\|_2^2$$

- p number of features
- n number of observations
- $X \in \mathbb{R}^{n \times pk}$ design matrix
- $Y \in \mathbb{R}^n$ response matrix
- $B \in \mathbb{R}^{pk}$ coefficient matrix
- K number of FPC scores
- $\|\cdot\|_2$ Frobenius norm for matrices
- $V \in \mathbb{R}^n$, $Z \in \mathbb{R}^{pk}$ dual variables
- $h^*(V)$, $\pi^*(Z)$ Fenchel-conjugate functions

Algorithm implementation

To find optimal *V, Z, B*

DAL method

(1) INNER SUB-PROBLEM: Given B^S, find V^{S+1} and Z^{S+1}:

$$(V^{s+1}, Z^{s+1}) \approx \arg\min_{V,Z} \mathcal{L}_\sigma(V, Z \mid B^s)$$

(2) UPDATE *B* and σ

$$B^{s+1} = B^s - \sigma^s\left(X^T V^{s+1} + Z^{s+1}\right)$$

$$\sigma^s \leq \sigma^{s+1} < \infty$$

Inner Sub-Problem

(●) UPDATE *V* and *Z* iteratively:

$$Z^{m+1} = \arg\min_Z \mathcal{L}_\sigma(Z, \mid V^{m+1}, B^s) = \mathbf{prox}_{\boldsymbol{\pi^*/\sigma}}\left(\boldsymbol{B^s/\sigma - X^T V^{m+1}}\right)$$

V^{m+1}: NO CLOSE SOLUTION

Algorithm implementation

To find optimal *V, Z, B*

DAL method

(1) INNER SUB-PROBLEM: Given B^S, find V^{S+1} and Z^{S+1}:

$$(V^{s+1}, Z^{s+1}) \approx \arg\min_{V,Z} \mathcal{L}_\sigma(V, Z \mid B^s)$$

(2) UPDATE *B* and σ

$$B^{s+1} = B^s - \sigma^s(X^T V^{s+1} + Z^{s+1})$$

$$\sigma^s \leq \sigma^{s+1} < \infty$$

Inner Sub-Problem

(●) UPDATE *V* and *Z* iteratively:

$$Z^{m+1} = \arg\min_Z \mathcal{L}_\sigma(Z, \mid V^{m+1}, B^s) = \mathbf{prox}_{\boldsymbol{\pi^*/\sigma}}\left(\boldsymbol{B^s/\sigma - X^T V^{m+1}}\right)$$

V^{m+1}: NO CLOSE SOLUTION

Semi-Smooth Newton Method

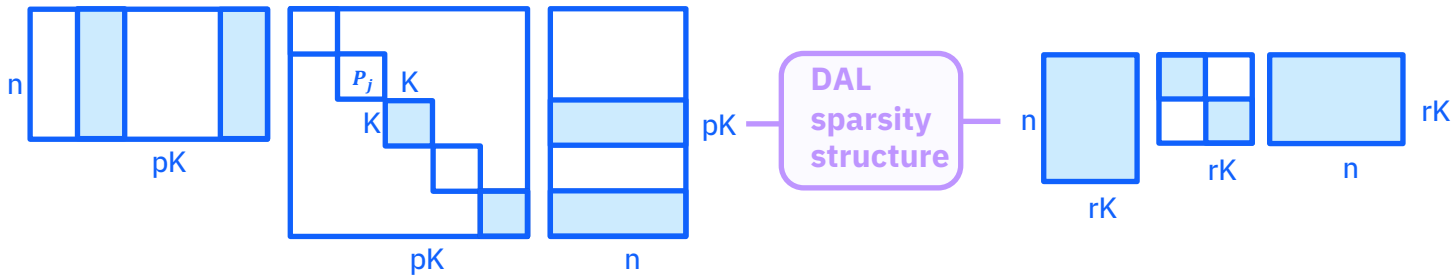
$$V^{m+1} = V^m + sD$$

With *s* step size and *D* descent direction:

$$\partial^2(\mathcal{L}(V))D = -\nabla\mathcal{L}(V)$$

Computational efficiency

$\partial^2(\mathcal{L}(V))$ has the following form:



Each $\boldsymbol{P_j}$ is associated with a feature

At each step just **r** active features

$\boldsymbol{P_j}$ is **0** for non-active features

TOTAL COST *(Cholesky + matrix multiplication):*

$$\mathcal{O}\left(pk(k^2 + pnk + p^2k^2 + n^2)\right) \longrightarrow \mathcal{O}\left(rk(k^2 + rnk + r^2k^2 + n^2)\right)$$

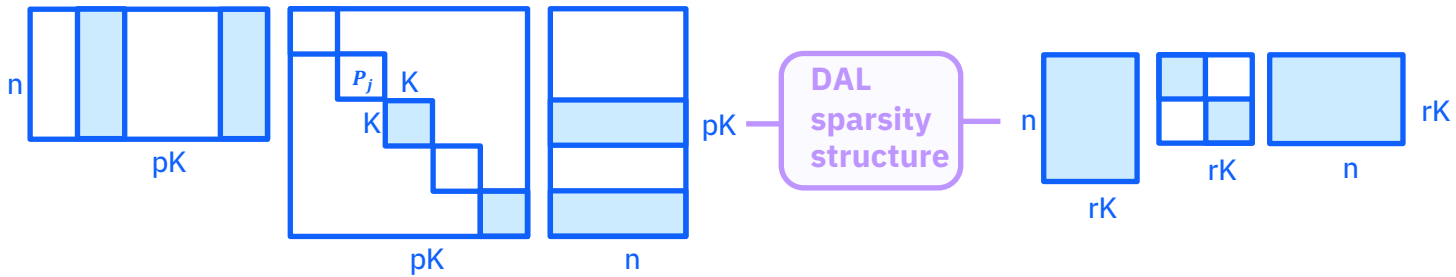
EXAMPLE:

In sparse settings $\mathbf{p \sim 10^5}$ after 1 iteration $\mathbf{r < 10^2}$

HUGE COMPUTATIONAL GAIN

Computational efficiency

$\partial^2(\mathcal{L}(V))$ has the following form:



Each $\boldsymbol{P_j}$ is associated with a feature

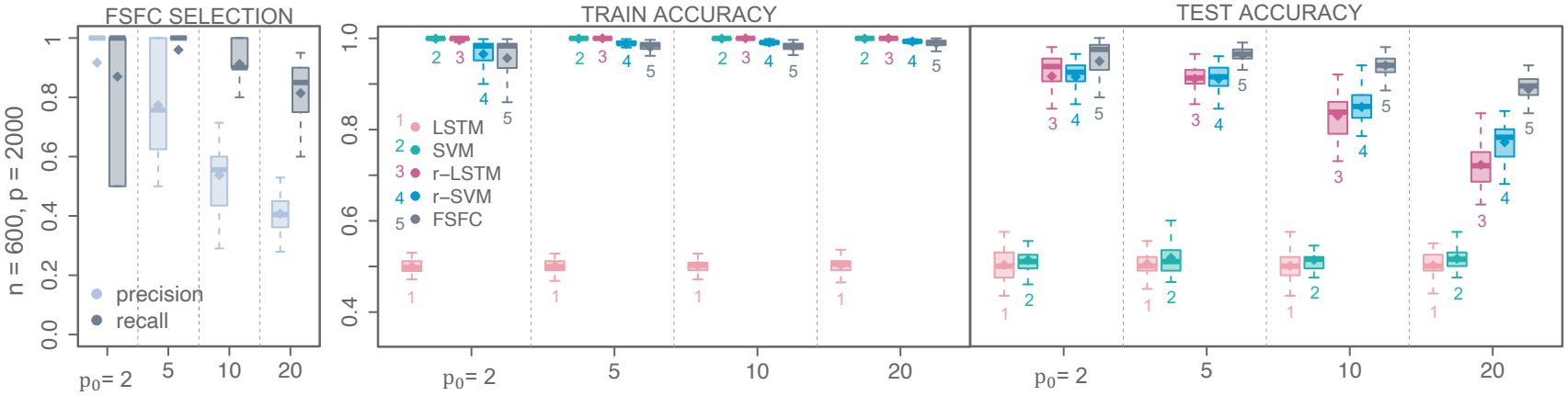
At each step just **r** active features

$\boldsymbol{P_j}$ is **0** for non-active features

	p_0	LSTM	SVM	FSFC	rLSTM	rSVM
$n = 300$ $p = 800$	2	140.61	16.17	1.44	5.56	0.01
	5	141.25	16.19	1.72	5.61	0.01
	10	139.93	16.17	1.84	5.66	0.02
	20	140.05	16.12	2.11	5.72	0.04
$n = 600$ $p = 2000$	2	355.66	144.18	5.44	9.52	0.01
	5	354.16	142.14	7.09	9.65	0.02
	10	348.51	141.96	7.58	9.86	0.05
	20	349.58	142.82	8.16	10.11	0.13

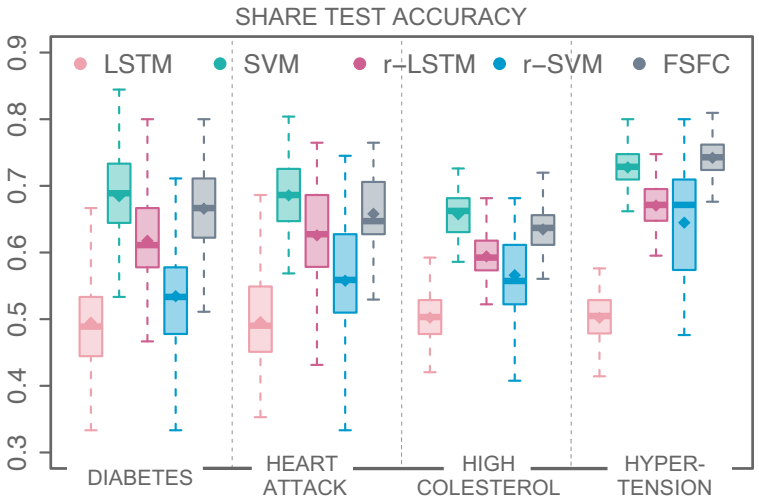
Time comparison (sec)

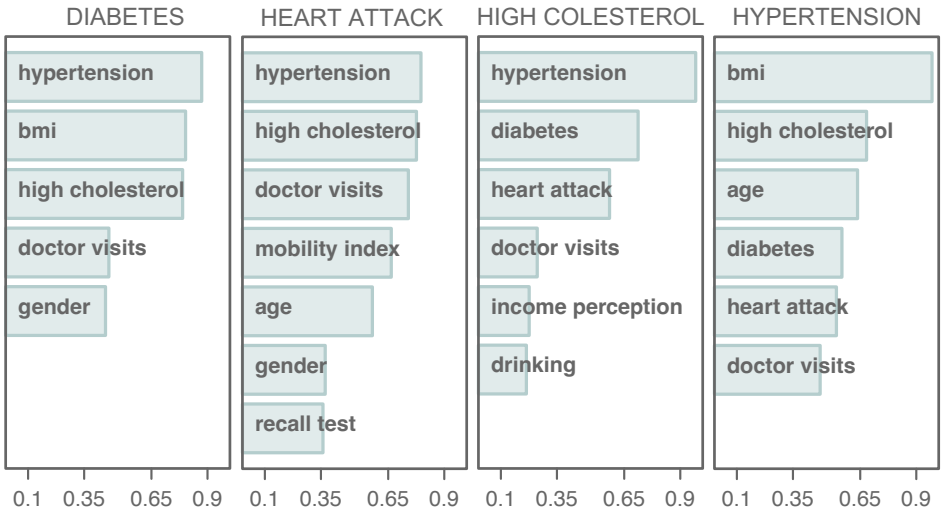
Simulation results



FSFC does **not overfit** the train, achieves the **best results**,
and **improves competitors' performances** when applied as a preprocessing step

SHARE data application





FSFC reveals **relationships** between **4 chronic diseases** and other **social** and **demographic factors**