Multivariate time series in healthcare: challenges and open questions

MulTiSA 2024: Workshop on Multivariate Time Series Analytics
In conjunction with ICDE 24

Laurent Oudre
laurent.oudre@ens-paris-saclay.fr

May, 13th 2024

Contents

The Centre Borelli

CENTRE BORELLI

The Centre Borelli
ENS Paris-Saclay, University Paris-Saclay, France

Fusion of two labs :

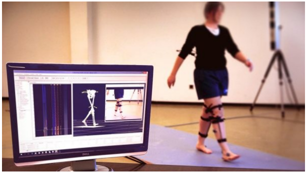
- The Centre de mathématiques et de leurs applications (CMLA) : applied mathematics for the study of complex phenomena and data
- The Cognition & Action Group (CognacG) : quantification and study of human and animal behavior

Main scientific questions

How to quantify human and animal behavior

- Adventure launched since 2012 : interdisciplinary collaboration between mathematicians, physicians, neuroscientists, engineers, biologists, etc...
- Implementation of measurement chains “pipelines”, platforms and intelligent tools but also of procedures for analysis, measurement and processing of data
- Creation of tools for diagnostic assistance, inter-individual comparison and longitudinal follow-up
- Integration into a clinical environment and interaction between algorithms and medical/neuroscience experts

First usecase : study of locomotion

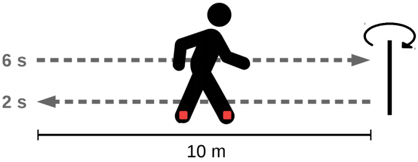


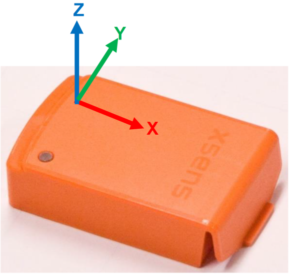
Study of locomotion

- Study the evolution of neurodegenerative diseases (Parkinson's, multiple sclerosis, etc.)
- Objective quantification of locomotion using inertial measurement units.
- Field testing: small wireless sensors and fully automatic device for routine use in medical consultations

First usecase: study of locomotion

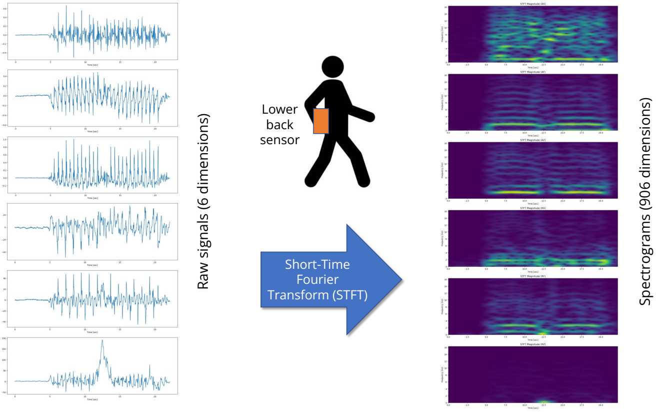
- Comfort speed protocol: stop (6 sec), walk forward (10 m), turn around, return, stop
- Four wireless inertial units: left foot, right foot, lower back, head
- Nine signals per sensor : linear acceleration (3D), angular velocity (3D), magnetic field (3D)





First usecase: study of locomotion

Lower back sensor, linear accelerations and angular velocities



These signals are not stationary: how can we find the exact temporal positions of each activity?

First usecase: study of locomotion

Study of locomotion: open questions linked to multivariation

- When processed in the time-frequency domain, the number of dimension is large ($d \approx 1000$). How can we discard useless dimensions to improve the processing step and reduce the computational time?
- How can we learn from annotated data which dimensions are relevant for the change-point detection task?

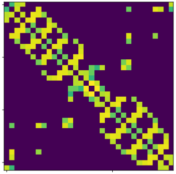
Supervised change-point detection

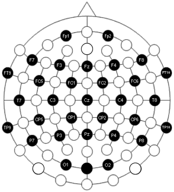
Use of a few annotated signals for the automatic calibration of change-point detection techniques based on dimension sparsity

Second usecase: study of general anesthesia

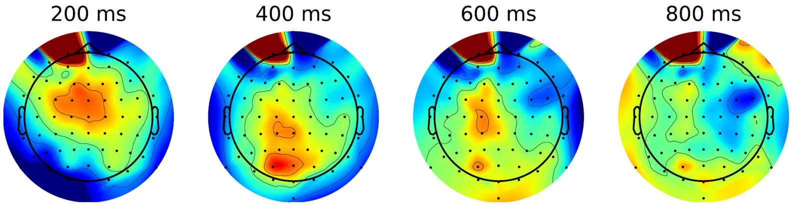
Study of general anesthesia

- Study of electroencephalography (EEG) signals recorded during general anesthesia in surgery context
- 32 sensors with sampling frequency 256 Hz





Second usecase: study of general anesthesia



- Presence of defective electrodes (due to the electronic scalpel or to bad contact between the electrode and the scalp)
- Yet, standard processing steps for EEG data include spatial filtering, where each signal value is replaced by a linear combination of all surrounding electrodes values
- One bad electrode: all the processing pipeline is flawed

Second usecase: study of general anesthesia

Study of general anesthesia: open questions linked to multivariation

- How can we learn the structure that allows to capture the redundancy of the data? What is the good representation domain for these data?
- How can we encode this stucture and use it to perform refined processing of the signals ?
- How can we construct an adapative filter that is robust to defective electrodes?

Graph signal processing

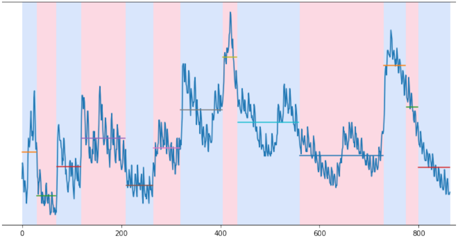
Use of graph learning and graph filtering techniques for the construction of adative spatial filters for EEG data

Outline of the presentation

- Two usecases with different approaches for handling/using/exploiting multivariation:
  - Study of locomotion: supervised change-point detection
  - Study of general anesthesia: graph approaches for handling multivariation
- Not focused on technical details but rather on the general scientific approaches
- Mostly preliminary works: open questions rather than closed answers :)

Contents

Problem statement: Change-Point Detection

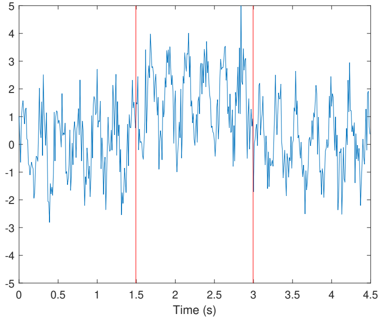


Change-Point Detection

Given a multivariate time series x$[t]$, retrieve the times $(t_1, \ldots, t_K)$ where a significant change occurs

- Necessitates to estimate both the change-points and the number of changes K
- Highly depends on the meaning given to *change*

Problem statement



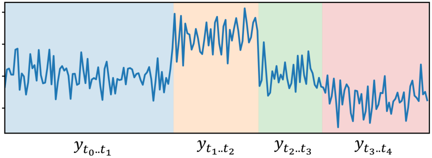
- Let assume that signal $\mathbf{x}[t]$ undergoes abrupt changes at times

$$\mathcal{T}^* = (t_1^*, \ldots, t_{K*}^*)$$

- Goal: retrieve the number of change-points K^* and their times $\mathcal{T}^*$
- Two assumptions: offline segmentation (but can easily be adapted to online setting) [Truong et al., 2020] and known number of changes K

Problem statement

$$(\hat{t}_1, \ldots, \hat{t}_K) = \underset{(t_1, \ldots, t_K)}{\operatorname{argmin}} \sum_{k=0}^{K} c(\mathbf{x}[t_k : t_{k+1}])$$

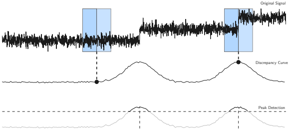


Cost function $c(.)$

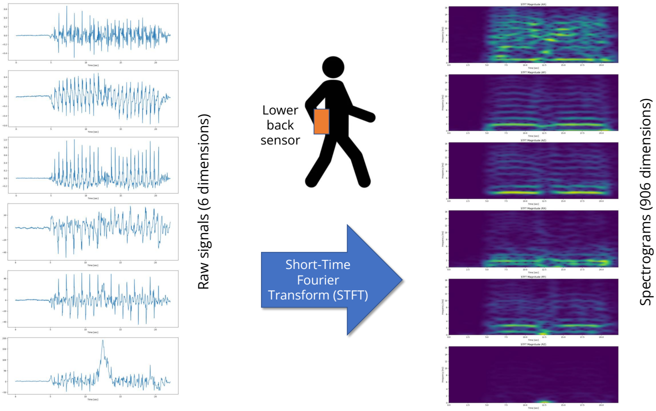
- Measures the homogeneity of the segments
- Choosing $c(.)$ conditions the type of change-points that we want to detect
- Often based on a probabilistic model for the data

Problem solving

- Optimal resolution in $\mathcal{O}(dKT^2)$ with dynamic programming
- Approximate resolution (sliding windows...)



Supervised change-point detection



Intuitively, all the dimensions are not useful: how can we select only the *good* dimensions where the changes occur?

Supervised change-point detection

- Supervised approach: we are going to learn an adequate cost function $c(.)$ that only selects some dimensions

$$c(\mathbf{x}[t_k : t_{k+1}]) = \sum_{t=t_k}^{t_{k+1}-1} \|\mathbf{x}_t - \bar{\mathbf{x}}[t_k : t_{k+1}]\|_{\mathbf{w}}$$

where $\|.\|_{\mathbf{w}}$ is a parametrized Mahalanobis-type (pseudo-)norm

$$\|\mathbf{y}\|_{\mathbf{w}} = \mathbf{y}^T \operatorname{diag}(\mathbf{w})\mathbf{y}$$

where $\operatorname{diag}(\mathbf{w})$ is a diagonal matrix for a vector $\mathbf{w} \in \mathbb{R}_+^d$ of positive weights.

- Calibration reduces to finding an appropriate $\mathbf{w}$, which can be seen as a scaling of each dimension p by w_p.

From annotations to triplets of samples

- Annotations are provided by an expert and transformed into triplet constraints, which are then fed to a sparse metric learning algorithm.
- For each training signal $\mathbf{y}^{(l)}$, a label consists in the set of change points $\mathcal{T}^{(l)} = \{t_1^{(l)}, t_2^{(l)}, \ldots\}$. The set $\mathcal{T}^{(l)}$ includes all the changes contained in the signal $y^{(l)}$, according to the expert.
- Using a label $\mathcal{T}^{(l)}$, a triplet can be created as follows
  - An anchor sample $\mathbf{y}_t$ in a certain segment $[t_k^{(l)}, t_{k+1}^{(l)}[$
  - A positive sample $\mathbf{y}_{t+}$ is any element of the same segment of $[t_k^{(l)}, t_{k+1}^{(l)}[$ (except the anchor sample)
  - A negative sample $\mathbf{y}_{t-}$ is any element of the previous segment $[t_{k-1}^{(l)}, t_k^{(l)}[$ or the following segment $[t_{k+1}^{(l)}, t_{k+2}^{(l)}[$

Metric learning

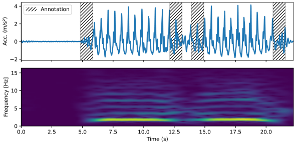
- Let $\mathcal{D}^{(l)}$ be the set of triplets generated from the labels. The sparse metric learning procedure for change-point detection consists in solving the following optimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^d_+}\left[\left(\sum_l \frac{1}{|\mathcal{D}^{(l)}|}\sum_{(y_t,y_{t^+},y_{t^-})\in\mathcal{D}^{(l)}}\left(1+\|y_t-y_{t^+}\|_w^2-\|y_t-y_{t^-}\|_w^2\right)_+\right)+\lambda\|w\|_1\right]$$

where $[\cdot]_+ = \max(0,\cdot)$ and $\lambda > 0$ controls the trade-off between the sparsity of w and the triplet constraints.

- Solving with uniform sampling of the label set + proximal stochastic gradient descent algorithm [Shi et al., 2014 ; De Vazelhes et al., 2020]

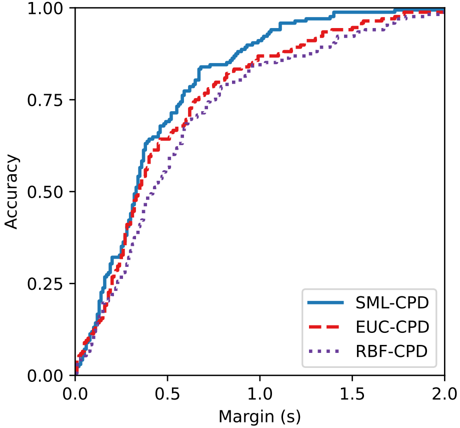
Data



- 42 labelled multivariate time series (sampling frequency: 100 Hz)
- Raw data from lower back sensor: $d = 6$ dimensions: the accelerations (m/s^2) along three axes (X, Y and Z) and the angular velocities (deg/s) around the same three axes.
- Time-frequency representation: Short-Term Fourier Transform (STFT). Concatenation of the 6 spectrograms yields a $d = 906$-dimensional signal.
- Comparison with 2 standard change-point detection methods (using Euclidean cost function or Gaussian kernel cost function), accuracy related to a error margin (from 0 to 2 seconds)
- 5-fold cross validation

Results

- Supervision improves detection accuracy. For margin $M = 1$ s, accuracies are 91.1% for *SML-CPD*, 86.9% for *EUC-CPD*, 83.9% for *RBF-CPD*.
- Our method projects the signals into a low dimension space. In the results, the number of non-zero coefficients in the learned $\mathbf{w}$ of *SML-CPD* is around 15 out of the 906 dimensions of the original STFT.
- Interpretability and useful insights on the segmentation.

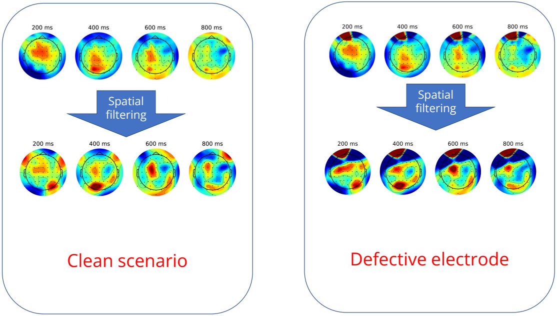


Open questions

- How far can we reduce the size of the training set?
- Could we learn more complex metrics?
- What if relevant dimensions change accross subjects?
- What if the changes are not synchronous between dimensions?

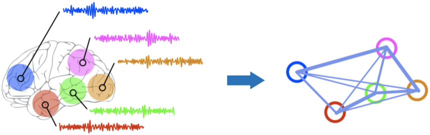
Contents

Construction of adaptive spatial filters for EEG data



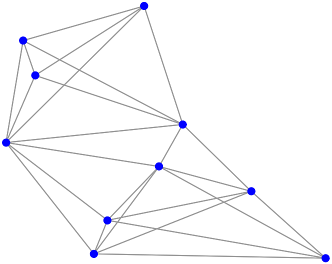
- Spatial filtering is a common step for EEG data: each electrode value is replaced by weighted combinations of signals from other electrodes
- Aims: increase topographical localization, facilitate electrode-level connectivity analyses
- Problem: very sensitive to noise and defective electrodes

Graph Signal Processing



- In most practical applications, the different dimensions of a multivariate signal $\mathbf{x}[t]$ are linked
  - Notion of correlation between recorded variables (ex: pressure/temperature/precipitation)
  - Sensor networks, body sensors, social networks...: spatial proximity, interactions...
- These links can be explicitly be modeled through a graph structure: **Graph Signal Processing** [Ortega et al., 2018]
- Each multivariate sample $\mathbf{x}[t]$ is assumed to be carried on the graph

What is a graph ?



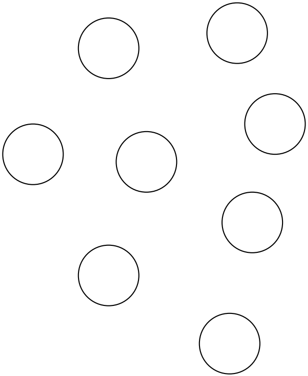
A graph G is a triplet $(\mathcal{V}, \mathcal{E}, \mathcal{W})$

- $\mathcal{V}$ is a finite set of d nodes (one per dimension, usually numbered $\{1, 2, ..., d\}$)
- $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ is a set of edges
- $\mathcal{W} : \mathcal{E} \rightarrow \mathbb{R}$ is a map from the set of edges to scalar values

Here : undirected graph, positive weights

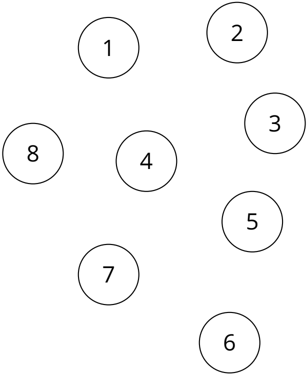
$\mathcal{W}(i, j)$ encodes the strength of the relationship between the dimensions i and j of the time series

Example



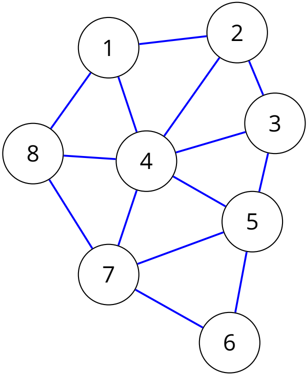
d nodes: one per signal dimension

Example



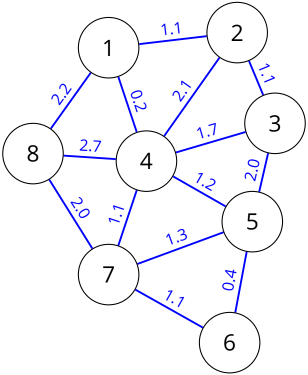
Dimension 1 lies on node 1, 2 on node 2, etc.

Example



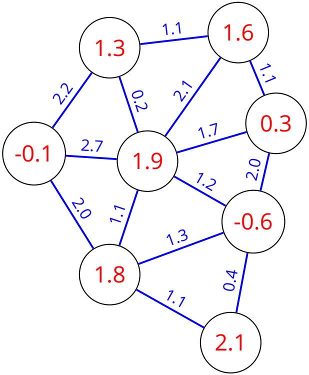
Edges model links between dimensions

Example



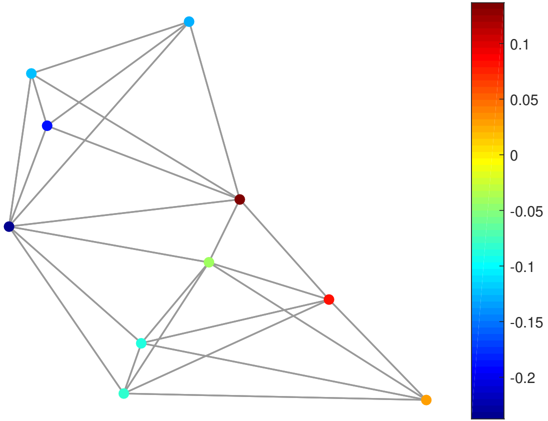
Weights model the strengths of these links

Example



Visualization of one multivariate sample $\mathbf{x}[t]$ on the graph

How to visualize graph signals?



Tasks

- Several machine learning tasks can be extended to graph signals [Ortega et al., 2018]:
  - Sampling/compression: choose the most relevant nodes (i.e. dimensions) to reconstruct the whole data
  - Graph inference: learn the graph structure from data [Mateos et al., 2019]
  - Denoising/filtering: use the graph structure to remove noise, outliers... [Chen et al., 2014]
  - Interpolation: use the graph structure to reconstruct missing data [Narang et al., 2013]
  - Classification, event detection, anomaly detection, prediction...
- Use the structure to improve performances of algorithms on multivariate time series

Outline of the approach

The method actually relies on the GSP framework for both steps

1. **Graph learning.** Learn the underlying graph structure modelling the links between the signal dimensions, by seeing samples $\mathbf{x}[0], \ldots, \mathbf{x}[N-1]$ as a collection of graph signals. [Humbert et al., 2021]
2. **Graph filtering.** Using the learned graph, design a spatial filter that adapts to the structure of the data [Humbert et al., 2021]

Two important notions: Laplacian of a graph + smoothness

Laplacian of a graph

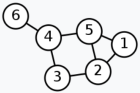
$$\mathbf{L} = \mathbf{D} - \mathbf{W}$$

- $\mathbf{W}$ (weight or adjacency matrix) :

$$W_{i,j} = \begin{cases} \mathcal{W}(i,j) & \text{if } (i,j) \in \mathcal{E} \\ 0 & \text{Otherwise} \end{cases}$$

- $\mathbf{D}$ (degree matrix) : diagonal matrix with

$$D_{i,i} = \sum_j W_{i,j}$$

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2&0&0&0&0&0\\ 0&3&0&0&0&0\\ 0&0&2&0&0&0\\ 0&0&0&3&0&0\\ 0&0&0&0&3&0\\ 0&0&0&0&0&1 \end{pmatrix}$	$\begin{pmatrix} 0&1&0&0&1&0\\ 1&0&1&0&1&0\\ 0&1&0&1&0&0\\ 0&0&1&0&1&1\\ 1&1&0&1&0&0\\ 0&0&0&1&0&0 \end{pmatrix}$	$\begin{pmatrix} 2&-1&0&0&-1&0\\ -1&3&-1&0&-1&0\\ 0&-1&2&-1&0&0\\ 0&0&-1&3&-1&-1\\ -1&-1&0&-1&3&0\\ 0&0&0&-1&0&1 \end{pmatrix}$

Symmetric, positive semi-definite

Laplacian of a graph

By construction of the Laplacian matrix

- $\forall \mathbf{x} \in \mathbb{R}^d, \quad \mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} W_{i,j} (x_i - x_j)^2 \geqslant 0$
- The constant vector $\mathbf{1}_d$ is an eigenvector for matrix $\mathbf{L}$ associated to eigenvalue $\lambda_1 = 0$
  Obvious as the sum of the matrix along the rows/column is equal to zero
- The number of connected components in the graph is equal to the number of eigenvalues equal to zero

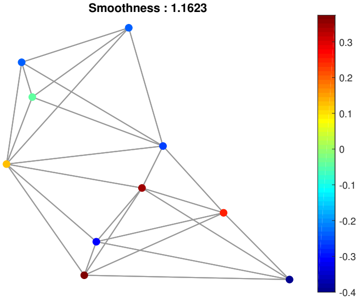
Smoothness

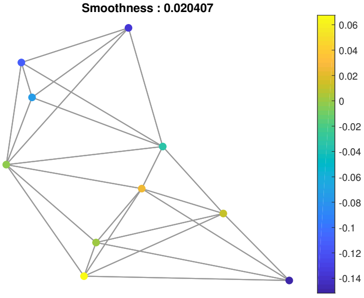
- Intuitively, signal values taken on adjacent nodes should be quite similar
- Notion of **smoothness** for a graph signal $\mathbf{x}$:

$$S(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x} = \frac{1}{2} \sum_{(i,j) \in \mathcal{E}} W_{i,j} (x_i - x_j)^2$$

- $S(\mathbf{x})$ is small if $(x_i - x_j)^2$ is small for large $W_{i,j}$
- Careful! This quantity in counterintuitive: large smoothness is achieved for non-smooth signals and vice-versa!

Example



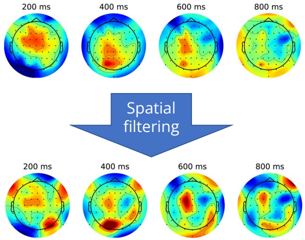


Smoothness decreases as the graph signal becomes more smooth

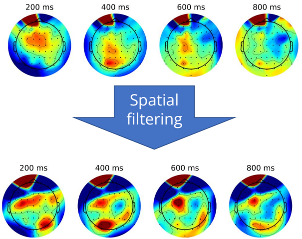
Back to the method

- Because of the volume conductivity problem, we need to remove parts of the signals which are common to all channels
- Ideas:
  - **Learn a graph structure on which the signals are as *smooth* as possible**
    Defective electrodes will probably be deconnected from others on the learned graph
  - **Remove the smooth approximation to the original signals**
    Defective electrodes that are not connected to others will not contribute

Results



Clean scenario



Defective electrode

Open questions

- What is a good graph structure?
- Could we model the time structure as a graph as well?
- Could we reconstruct the corrupted data by using the graph structure?

Contents

Conclusion

- All the problems encountered in this talk (non-stationarity, noise) are probably due to the *real-life* setting and to the fact that measurements are done out of the lab
- Multivariation can bring redundancy and be a benefit for data processing... or, when useless/bad dimensions are present, dramatically increase the difficulty of the problems!
- More about this during the panel session :-)

References

- Part I: Change-point detection
  - Overview of change-point detection
    ⟶ C. Truong, L. Oudre, and N. Vayatis. Selective review of offline change point detection methods. Signal Processing, 167:107299, 2020
  - Supervised change-point detection
    ⟶ C. Truong, and L. Oudre. Supervised change-point detection with dimension reduction. In Proceedings of the European Signal Processing Conference (EUSIPCO), pages 1005-1009, Helsinki, Finland, 2023
    ⟶ C. Truong, L. Oudre, and N. Vayatis. Supervised kernel change point detection with partial annotations. In Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pages 3147-3151, Brighton, UK, 2019
- Part II: Graph Signal Processing (GSP)
  - Overview of GSP
    ⟶ A. Ortega, P. Frossard, J. Kovačević, J.M. Moura, and P. Vandergheynst Graph signal processing: Overview, challenges, and applications. In Proceedings of the IEEE, 106(5), 808-828, 2018
  - Graph learning
    ⟶ P. Humbert, B. Le Bars, L. Oudre, A. Kalogeratos, and N. Vayatis. Learning Laplacian Matrix from Graph Signals with Sparse Spectral Representation. Journal of Machine Learning Research, 22(195):1-47, 2021
  - Application to EEG data
    ⟶ P. Humbert, L. Oudre, and C. Dubost. Learning spatial filters from EEG signals with Graph Signal Processing methods. In Proceedings of the International Conference of the IEEE Engineering in Medecine and Biology Society (EMBC), pages 657-660, Guadalajara, Mexico, 2021
- Gait Datasets
  ⟶ C. Voisard, N. De l'Escalopier, A. Moreau, A. Vienne-Jumeau, D. Ricard, and L. Oudre. A Reference Data Set for the Study of Healthy Subject Gait with Inertial Measurements Units. Image Processing On Line, 13:314-320, 2023
  ⟶ C. Truong, R. Barrois-Müller, T. Moreau, C. Provost, A. Vienne-Jumeau, A. Moreau, P.-P. Vidal, N. Vayatis, S. Buffat, A. Yelnik, D. Ricard, and L. Oudre. A Data Set for the Study of Human Locomotion with Inertial Measurements Units. Image Processing On Line, 9:381-390, 2019

Questions?

Additional slides

Graph Fourier Transform

Given a signal $\mathcal{G}$ with only one connex component, we compute the eigen-decomposition of its Laplacian $\mathbf{L}$:

$$\mathbf{L} = \mathbf{U} \begin{bmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_N \end{bmatrix} \mathbf{U}^T, \quad 0 = \lambda_1 < \lambda_2 \leqslant ... \leqslant \lambda_N$$

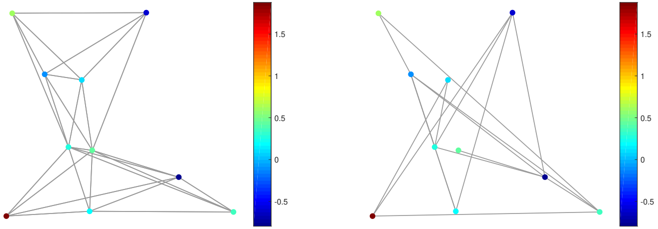
- λ_i are interpretable as **frequencies**: smoothness of eigenvector $\mathbf{u}_i$
- $\mathbf{u}_i$ is the eigenvector associated to frequency λ_i
- Decomposition on a basis of graph signals of decreasing smoothness

Graph Fourier Transform

The Graph Fourier Transform (GFT) $\hat{\mathbf{x}}$ of a graph signal $\mathbf{x} \in \mathbb{R}^N$ is defined as

$$\hat{\mathbf{x}} = \mathbf{U}^T \mathbf{x}$$

Graph inference



- **Aim :** Given a collection of n observed graph signals $\{\mathbf{y}^{(k)}\}_{k=1}^{n}$ of size N, learn the graph $\mathcal{G}$ that best explains the structure observed in the signals
- **Assumption :** The graph signals $\mathbf{Y}$ should be bandlimited and smooth for $\mathcal{G}$
- **Inputs :**

$$\mathbf{Y} = [\mathbf{y}^{(1)}, \cdots, \mathbf{y}^{(n)}] \in \mathbb{R}^{N \times n} : \text{input graph signals}$$

- **Outputs :**

$$\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T : \text{Laplacian matrix of } \mathcal{G}$$

$$\hat{\mathbf{Y}} : \text{GFT of signals } \mathbf{Y} \text{ on } \mathcal{G}$$

Problem formulation

$$\min_{\hat{\mathbf{Y}},\mathbf{U},\Lambda} \|\mathbf{Y} - \mathbf{U}\hat{\mathbf{Y}}\|_F^2 + \alpha\|\Lambda^{1/2}\hat{\mathbf{Y}}\|_F^2 + \beta\|\hat{\mathbf{Y}}\|_S$$

$$\text{s.t.} \quad \begin{cases} \mathbf{U}^T\mathbf{U} = \mathbf{I}_N,\ \mathbf{u}_1 = \frac{1}{\sqrt{N}}\mathbf{1}_N, & \text{(a)} \\ (\mathbf{U}\Lambda\mathbf{U}^T)_{k,l} \leqslant 0 \quad k \neq l, & \text{(b)} \\ \Lambda = \mathrm{diag}(0, \lambda_2, \ldots, \lambda_N) \succeq 0, & \text{(c)} \\ \mathrm{tr}(\Lambda) = N \in \mathbb{R}_*^+. & \text{(d)} \end{cases}$$

- $\|\mathbf{Y} - \mathbf{U}\hat{\mathbf{Y}}\|_F^2$: $\mathbf{Y}$ should be close to the inverse GFT of its spectral representation in $\mathcal{G}$
- $\|\Lambda^{1/2}\hat{\mathbf{Y}}\|_F^2$: $\mathbf{Y}$ should be smooth on $\mathcal{G}$
- $\|\hat{\mathbf{Y}}\|_S$: sparsity constraint on the frequency representation of Y

Constraints : $\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T$ should be a Laplacian matrix (symmetric, semi positive)

Resolution with alternate minimization

Graph filtering

- The filtered signal is obtained by solving the following optimization problem:

$$\mathbf{y}_F = \arg\min_{\mathbf{s}} \|\mathbf{y} - \mathbf{s}\|_2^2 + \gamma \mathbf{s}^t \hat{\mathbf{L}} \mathbf{s} \ . \quad (1)$$

- The *smooth* component $\mathbf{y}_F$ is then removed from the signals