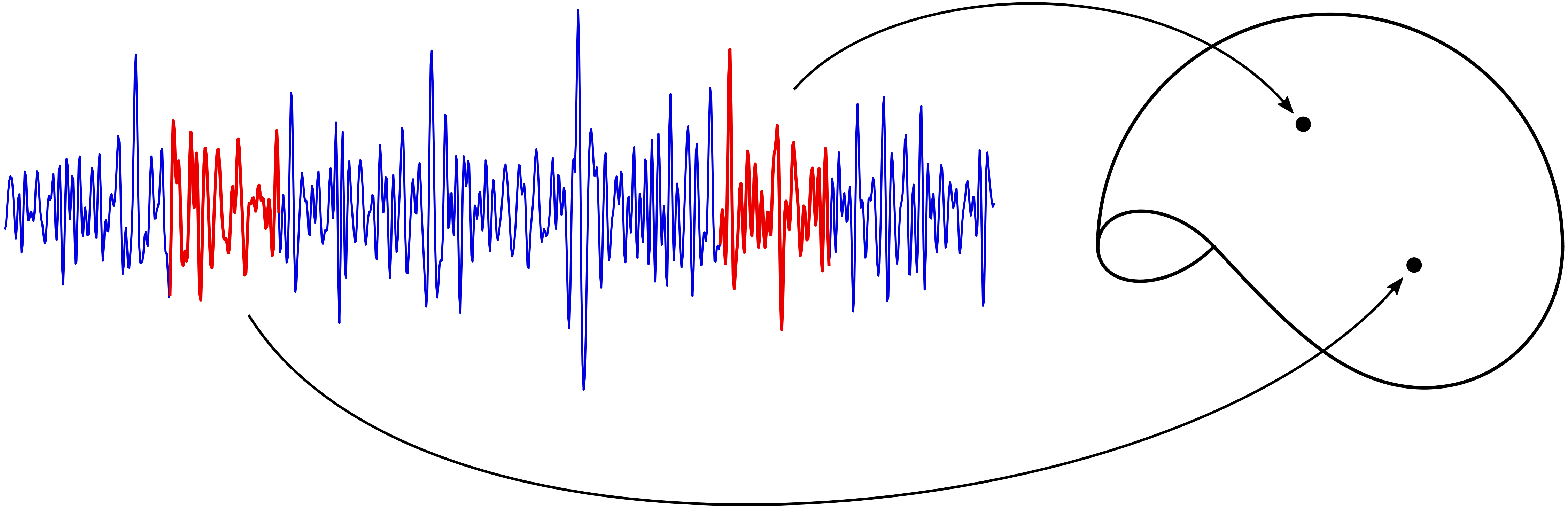
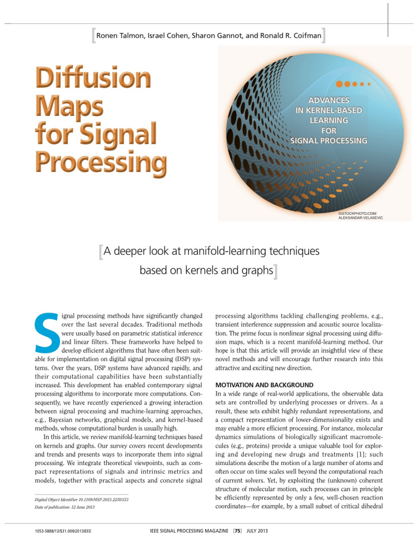
An Introduction to Manifold Learning

and its use in time series analysis

Pedro L. C. Rodrigues – Machine Learning Seekers of Truth – 6th December 2017





[Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald R. Coifman]

Diffusion Maps for Signal Processing

ADVANCES IN KERNEL-BASED LEARNING FOR SIGNAL PROCESSING

©ISTOCKPHOTO.COM/ ALEKSANDAR VELASEVIC

[A deeper look at manifold-learning techniques based on kernels and graphs]

Signal processing methods have significantly changed over the last several decades. Traditional methods were usually based on parametric statistical inference and linear filters. These frameworks have helped to develop efficient algorithms that have often been suitable for implementation on digital signal processing (DSP) systems. Over the years, DSP systems have advanced rapidly, and their computational capabilities have been substantially increased. This development has enabled contemporary signal processing algorithms to incorporate more computations. Consequently, we have recently experienced a growing interaction between signal processing and machine-learning approaches, e.g., Bayesian networks, graphical models, and kernel-based methods, whose computational burden is usually high.

In this article, we review manifold-learning techniques based on kernels and graphs. Our survey covers recent developments and trends and presents ways to incorporate them into signal processing. We integrate theoretical viewpoints, such as compact representations of signals and intrinsic metrics and models, together with practical aspects and concrete signal processing algorithms tackling challenging problems, e.g., transient interference suppression and acoustic source localization. The prime focus is nonlinear signal processing using diffusion maps, which is a recent manifold-learning method. Our hope is that this article will provide an insightful view of these novel methods and will encourage further research into this attractive and exciting new direction.

MOTIVATION AND BACKGROUND

In a wide range of real-world applications, the observable data sets are controlled by underlying processes or drivers. As a result, these sets exhibit highly redundant representations, and a compact representation of lower-dimensionality exists and may enable a more efficient processing. For instance, molecular dynamics simulations of biologically significant macromolecules (e.g., proteins) provide a unique valuable tool for exploring and developing new drugs and treatments [1]; such simulations describe the motion of a large number of atoms and often occur on time scales well beyond the computational reach of current solvers. Yet, by exploiting the (unknown) coherent structure of molecular motion, such processes can in principle be efficiently represented by only a few, well-chosen reaction coordinates—for example, by a small subset of critical dihedral

Digital Object Identifier 10.1109/MSP.2013.2250353
Date of publication: 12 June 2013

1053-5888/13/$31.00©2013IEEE IEEE SIGNAL PROCESSING MAGAZINE [75] JULY 2013

Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald Coifman (2013)
Diffusion maps for signal processing: A deeper look at manifold-learning techniques based on kernels and graphs

Stéphane Lafon (Ph.D. thesis, 2004)
Diffusion maps and Geometric Harmonics

Ronald Coifman and Stéphane Lafon (2006)
Diffusion maps

Outline

Introduction

Manifold Learning for Time-Series Analysis

Some results

Concluding remarks

Outline

Introduction

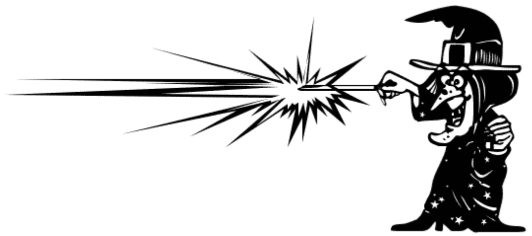
Manifold Learning for Time-Series Analysis

Some results

Concluding remarks

High-dimensionality is an **obstacle** to efficient data processing for a number of reasons :

- Norms in $\mathbf{R}^n$ are not **numerically** equivalent when n is large
- Density estimation is **difficult** in big dimensions (needs to much data)
- Fast algorithms may become **prohibitively slow** in high dimensions

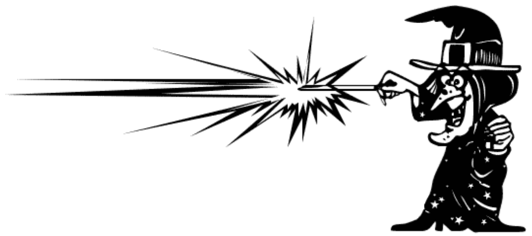


curse of dimensionality

gipsa-lab

High-dimensionality is an **obstacle** to efficient data processing for a number of reasons :

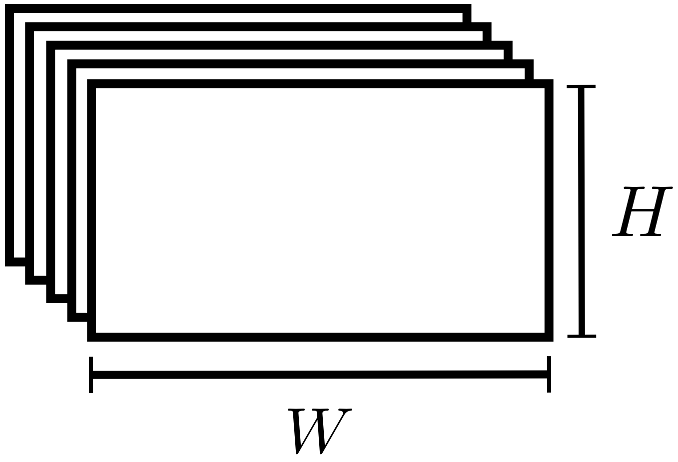
- Norms in $\mathbf{R}^n$ are not **numerically** equivalent when n is large
- Density estimation is **difficult** in big dimensions (needs to much data)
- Fast algorithms may become **prohibitively slow** in high dimensions



curse of dimensionality

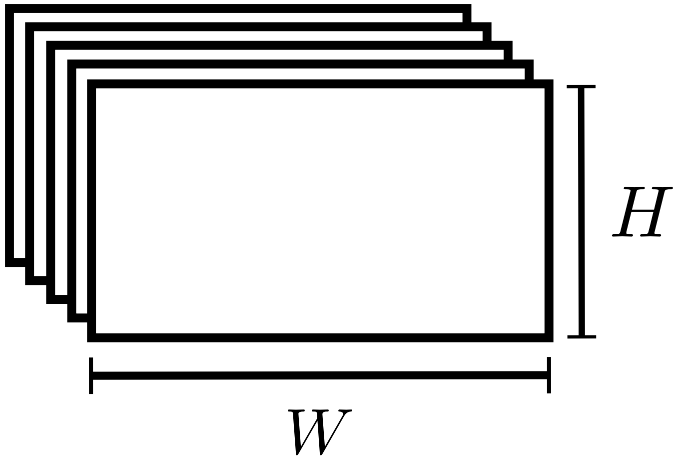
Data dimensions are often correlated via some **functional dependence**

Fundamental assumption : data lies on (or close to) a low-dimensional manifold



A video is composed of several **frames** $\left\{X_i\right\}_{i=1}^{N}$

Each frame is an image and can be associated to a **vector**



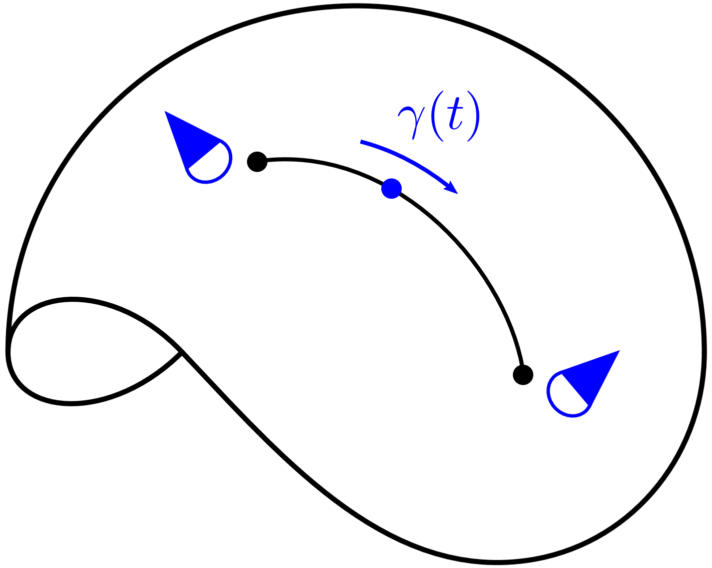
A video is composed of several **frames** $\left\{X_i\right\}_{i=1}^{N}$

Each frame is an image and can be associated to a **vector**

Frames are points in a **very** high-dimensional space,

but there are not many **degrees of freedom** in the video

$$X_i \in \mathbb{R}^{W \times H} \Rightarrow \Phi\left(X_i\right) \in \mathbb{R}^d$$



- Classical techniques likes PCA and MDS for **linear** dimensionality reduction
- Two papers in Science magazine (2000) presenting **Manifold Learning** : Isomap and LLE
- Previous methods are all **global** (except LLE)
- Diffusion maps and Laplacian Eigenmaps search for an embedding that respects **local** geometry

Think globally, act locally

Dimensionality Reduction for Multivariate Time-Series

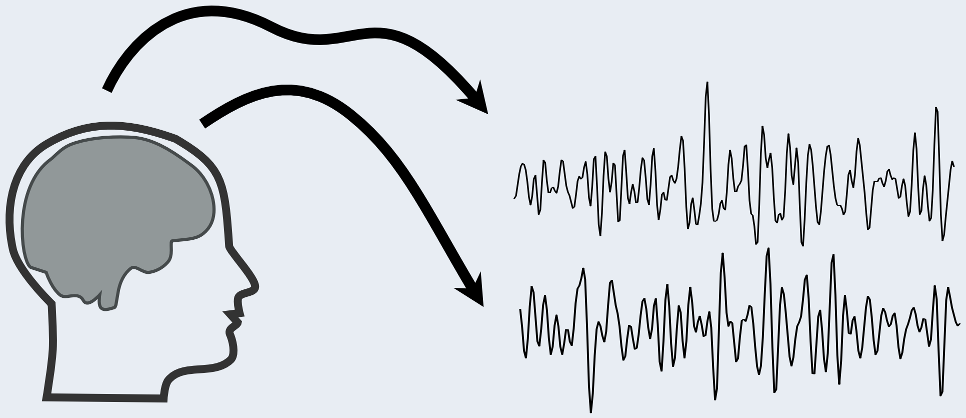
Multivariate time-series often present **redundancy** among its dimensions

$$\boldsymbol{x}(n) = \begin{bmatrix} x_1(n) \\ \vdots \\ x_N(n) \end{bmatrix}$$

Dimensionality Reduction for Multivariate Time-Series

Multivariate time-series often present **redundancy** among its dimensions

$$\boldsymbol{x}(n) = \begin{bmatrix} x_1(n) \\ \vdots \\ x_N(n) \end{bmatrix}$$



Cognitive task
(*K* trials)

EEG recording

$$\{X_1, X_2, \ldots, X_K\}$$

$$X_i \in \mathbb{R}^{N \times T}$$

N – electrodes
T – samples

Dimensionality Reduction for Multivariate Time-Series

Multivariate time-series often present **redundancy** among its dimensions

$$\boldsymbol{x}(n) = \begin{bmatrix} x_1(n) \\ \vdots \\ x_N(n) \end{bmatrix}$$

REDUNDANCY → **COMPRESSION**

$$\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$$

→

$$\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^{d}$$

Two main categories of dimensionality reduction : **linear** and **non-linear** approaches

gipsa-lab

Dimensionality Reduction for Multivariate Time-Series

Spectral Embedding – Manifold Learning

Non-linear methods based on **three** basic steps

(Laplacian Eigenmaps, Diffusion Maps, Kernel PCA, etc.)

(1) Construct a **distance** matrix

(2) Define a **Laplacian** matrix

(3) Spectral **decomposition** of L

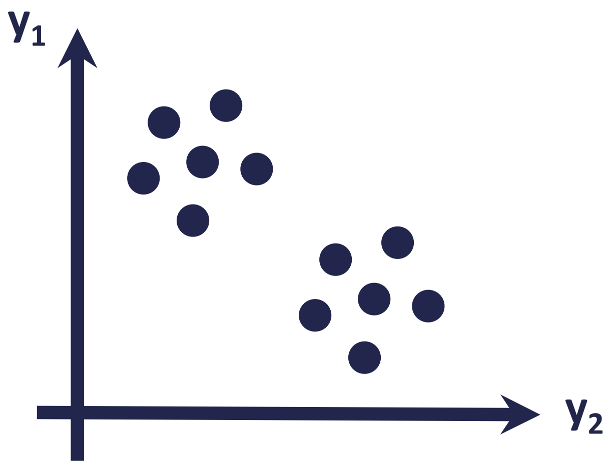
The distance matrix can be interpreted as defining a **weighted graph** where the **nodes** are the trials and the **weights** their distances

gipsa-lab

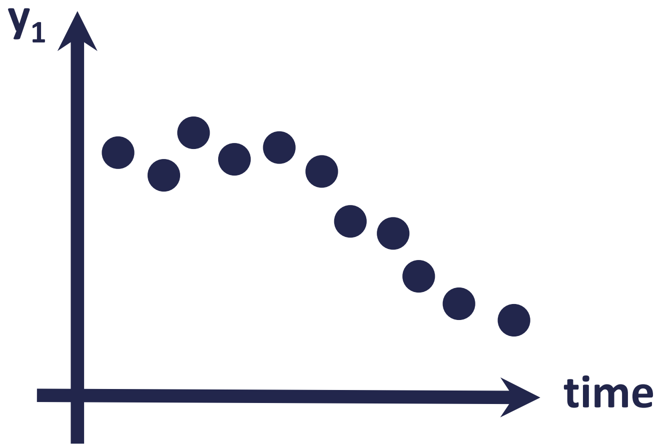
What can we do with this ? Unsupervised time-series analysis

$$\{X_1, X_2, \ldots, X_K\}$$
$$X_i \in \mathbb{R}^{N \times T}$$

are **recordings** of different trials



are samples of a **sliding-window**

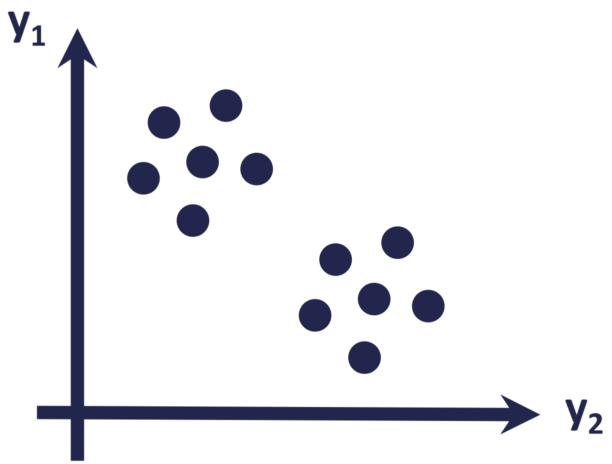


gipsa-lab

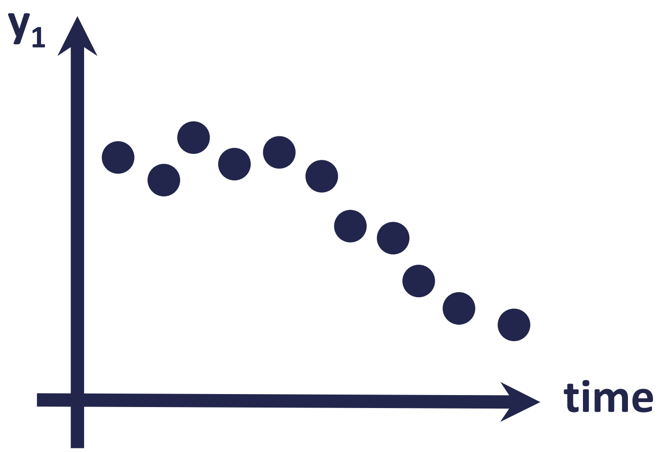
What can we do with this ? Unsupervised time-series analysis

$$\{X_1, X_2, \ldots, X_K\}$$
$$X_i \in \mathbb{R}^{N \times T}$$

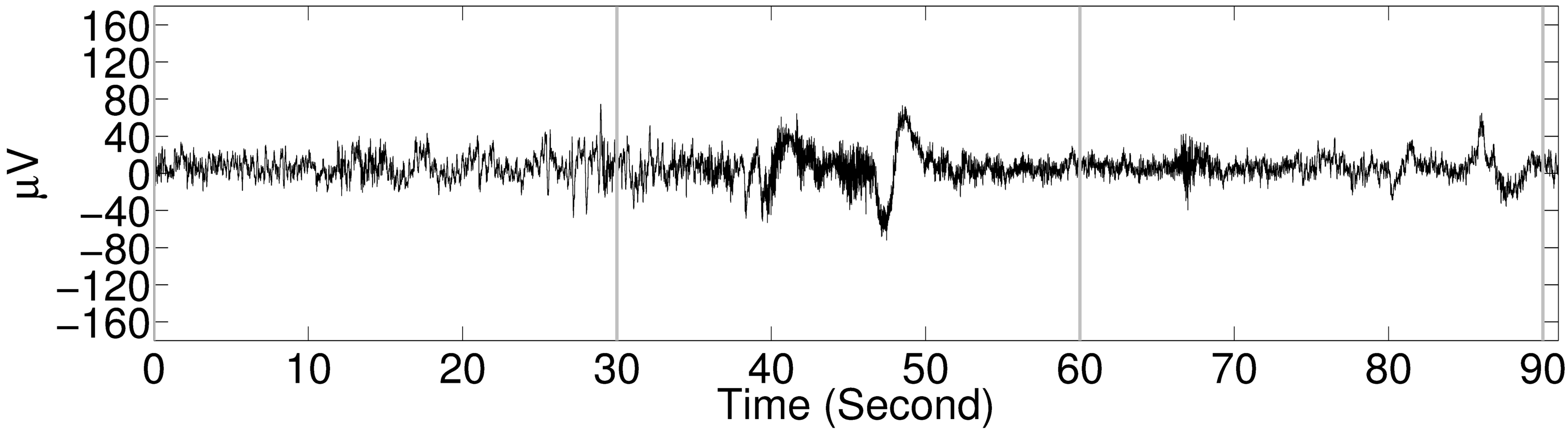
are **recordings** of different trials

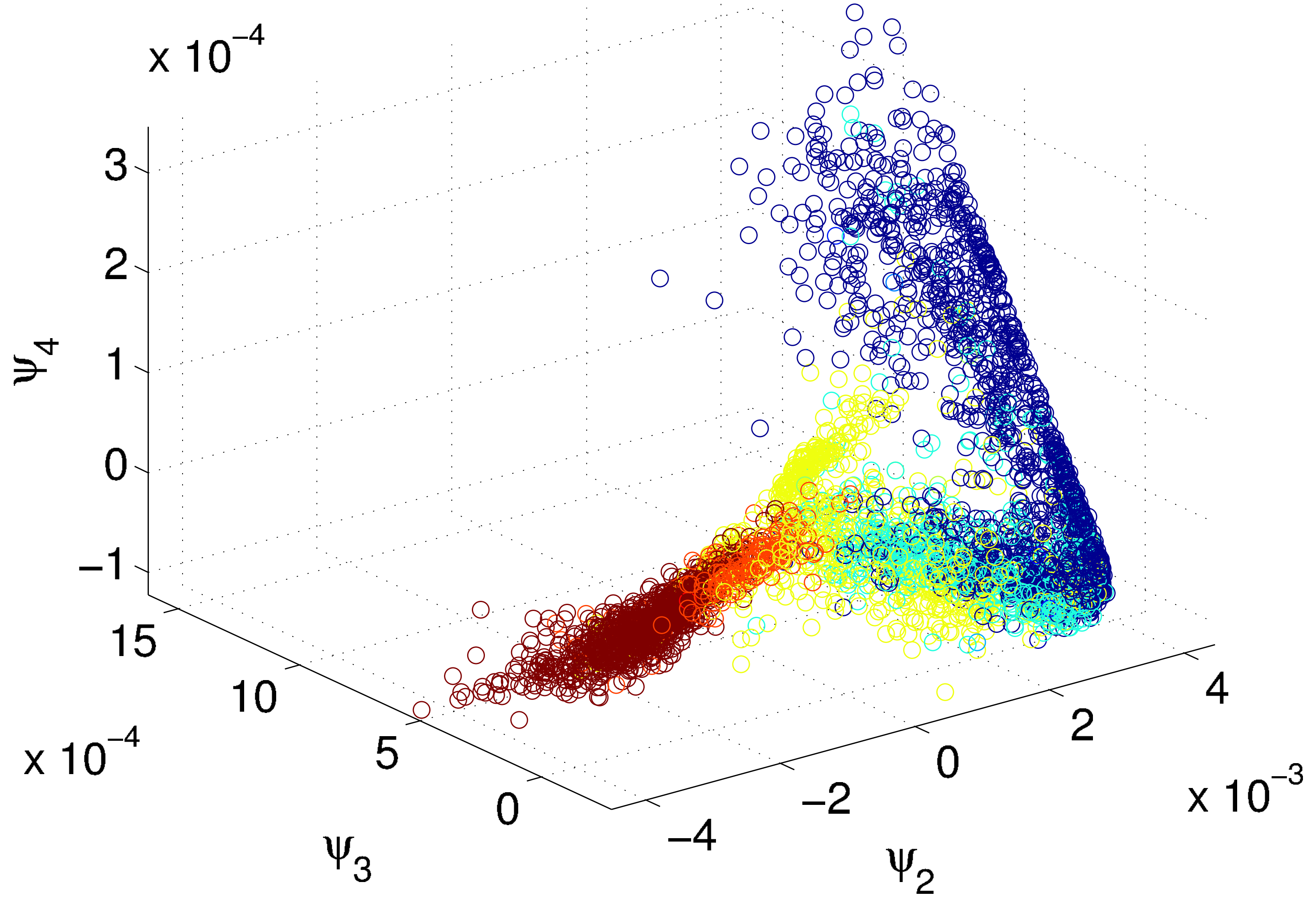


are samples of a **sliding-window**



Hau-tieng Wu, Ronen Talmon, and Yu-Lun Lo
Assess sleep stage by modern signal processing techniques
IEEE Transactions on Biomedical Engineering (2015)




x 10^−4
3
2
1
0
−1
ψ4
15
10
5
0
x 10^−4
ψ3
−4
−2
0
2
4
x 10^−3
ψ2

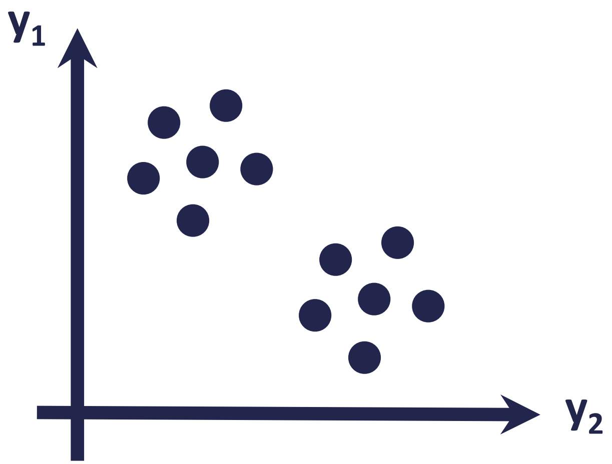
What can we do with this ? Unsupervised time-series analysis

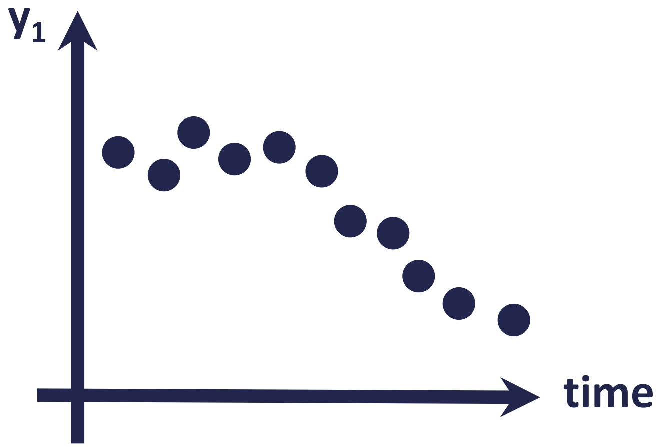
are **recordings** of different trials

$$\{X_1, X_2, \dots, X_K\}$$

$$X_i \in \mathbb{R}^{N \times T}$$

are samples of a **sliding-window**

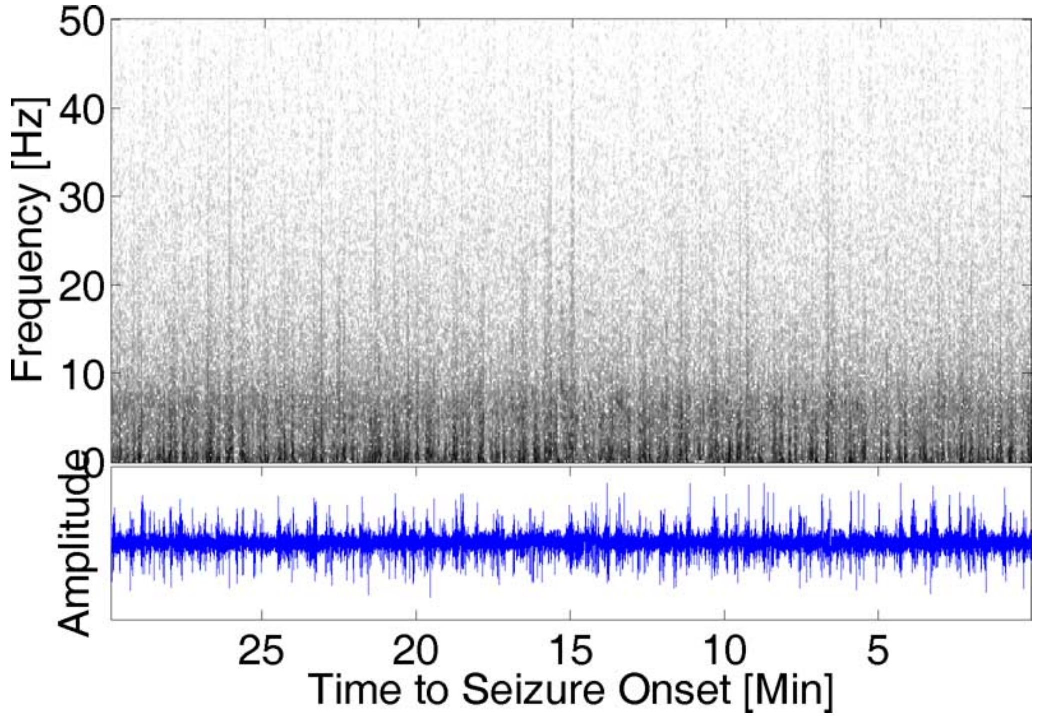


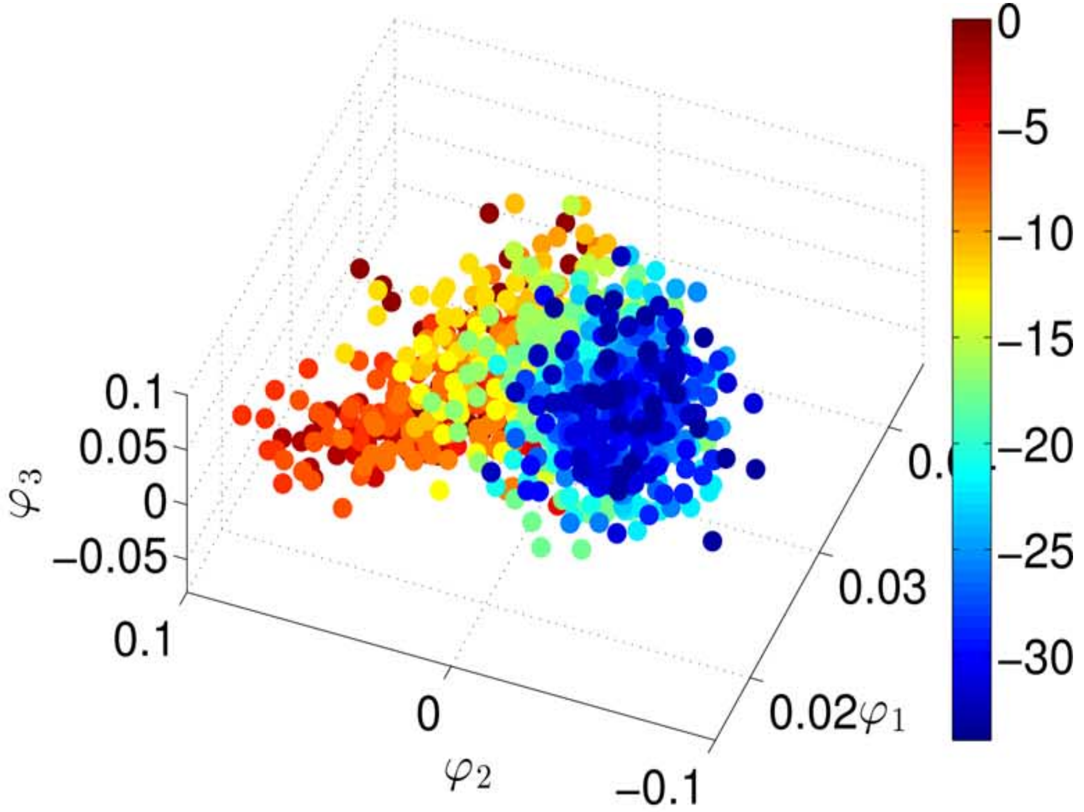


Ronen Talmon, Stéphane Mallat, Hitten Zavari, Ronald R. Coifman

Manifold Learning for Latent Variable Inference in Dynamical Systems

IEEE Transactions in Signal Processing (2015)





Outline

Introduction

Manifold Learning for Time-Series Analysis

Some results

Concluding remarks

Spectral embedding of Multivariate Time-Series

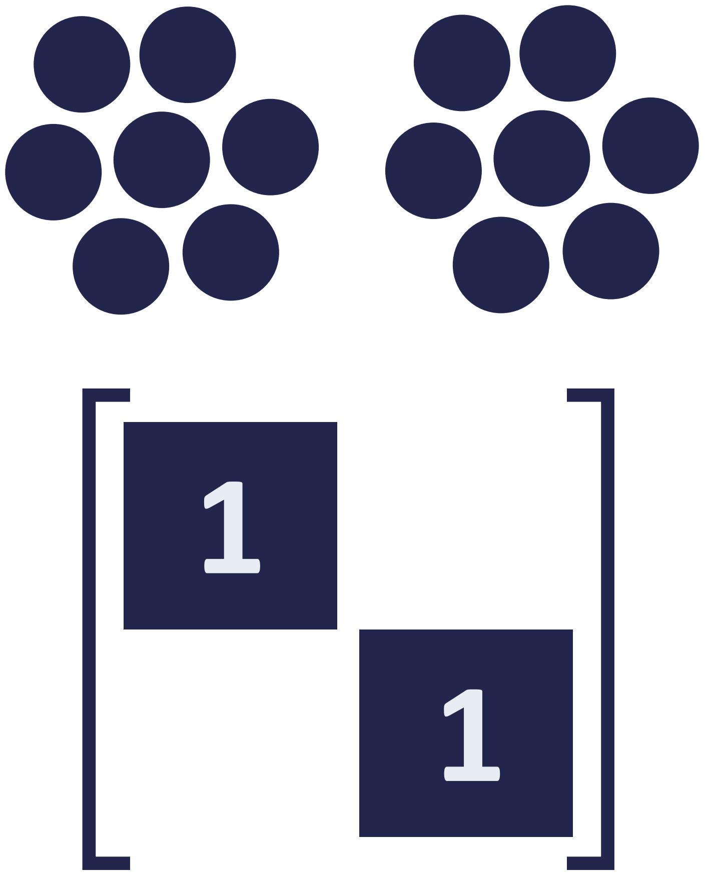
Construct a pairwise **distance matrix** between the different trials :

$$\{X_1, X_2, \dots, X_K\} \qquad \mathbf{D}_{ij} = d(X_i, X_j)$$

and use it to define an **affinity matrix** via a Gaussian kernel :

$$\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$$

There are **several** heuristics for the affinity matrix (k-NN, thresh.)



Spectral embedding of Multivariate Time-Series

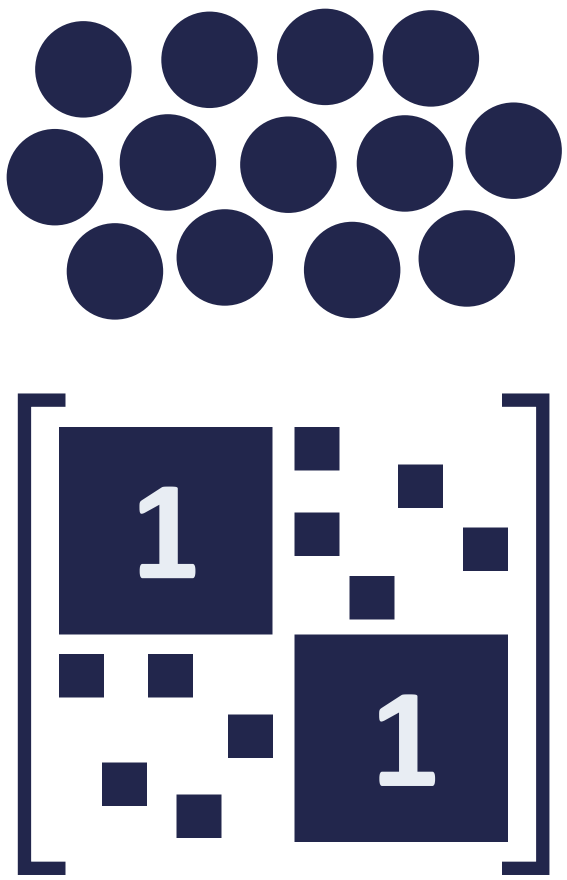
Construct a pairwise **distance matrix** between the different trials :

$$\{X_1, X_2, \ldots, X_K\} \qquad \mathbf{D}_{ij} = d(X_i, X_j)$$

and use it to define an **affinity matrix** via a Gaussian kernel :

$$\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$$

There are **several** heuristics for the affinity matrix (k-NN, thresh.)



Spectral embedding of Multivariate Time-Series

The embedding is done so to **minimize** a cost function involving the affinities between points

$$\mathcal{L} = \frac{1}{2} \sum_{i,j} (y_i - y_j)^2 \; \mathbf{W}_{ij}$$

and a normalization $\sum_i \mathbf{D}_{ii} \; y_i^2 = 1$ where $\mathbf{D}_{ii} = \sum_j \mathbf{W}_{ij}$

$\{X_1, X_2, \ldots, X_K\}$

original points

$\{y_1, y_2, \ldots, y_K\}$

embedded points

Spectral embedding of Multivariate Time-Series

Rewriting the terms and defining the **Laplacian** matrix as $\mathbf{L} = \mathbf{D} - \mathbf{W}$ we have that

$$\mathcal{L} = \frac{1}{2}\sum_{i,j}(y_i - y_j)^2\ \mathbf{W}_{ij} = \mathbf{y}^T\mathbf{L}\mathbf{y} \quad \text{with} \quad \mathbf{y} = \begin{bmatrix} y_1 & \dots & y_K \end{bmatrix}^T$$

The minimization of the cost function can be recast as a **generalized** eigenvalue problem

$$\begin{aligned} &\text{minimize} && \mathbf{y}^T\mathbf{L}\mathbf{y} \\ &\text{s.t.} && \mathbf{y}^T\mathbf{D}\mathbf{y} = 1 \end{aligned}$$

The embedded coordinates are the elements of the eigenvectors of the Laplacian matrix

Summing up

- $\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$ **Gather** recordings on different trials

- $\mathbf{D}_{ij} = d(X_i, X_j)$

- $\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$

- $\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^d$

Summing up

- $\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$
- $\mathbf{D}_{ij} = d(X_i, X_j)$ Define a notion of **distance** between trials
- $\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$
- $\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^d$

Summing up

- [ ] $\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$
- [ ] $\mathbf{D}_{ij} = d(X_i, X_j)$
- [x] $\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$ **Construct** the affinity and Laplacian matrices

  (Laplacian Eingenmaps and Diffusion Maps define the affinity matrix differently)
- [ ] $\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^d$

Summing up

- [ ] $\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$
- [ ] $\mathbf{D}_{ij} = d(X_i, X_j)$
- [ ] $\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$
- [x] $\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^d$ Get the spectral **embedding**

$$Y_i = \begin{bmatrix} \mathbf{y}_1(i) \\ \vdots \\ \mathbf{y}_d(i) \end{bmatrix}$$

where

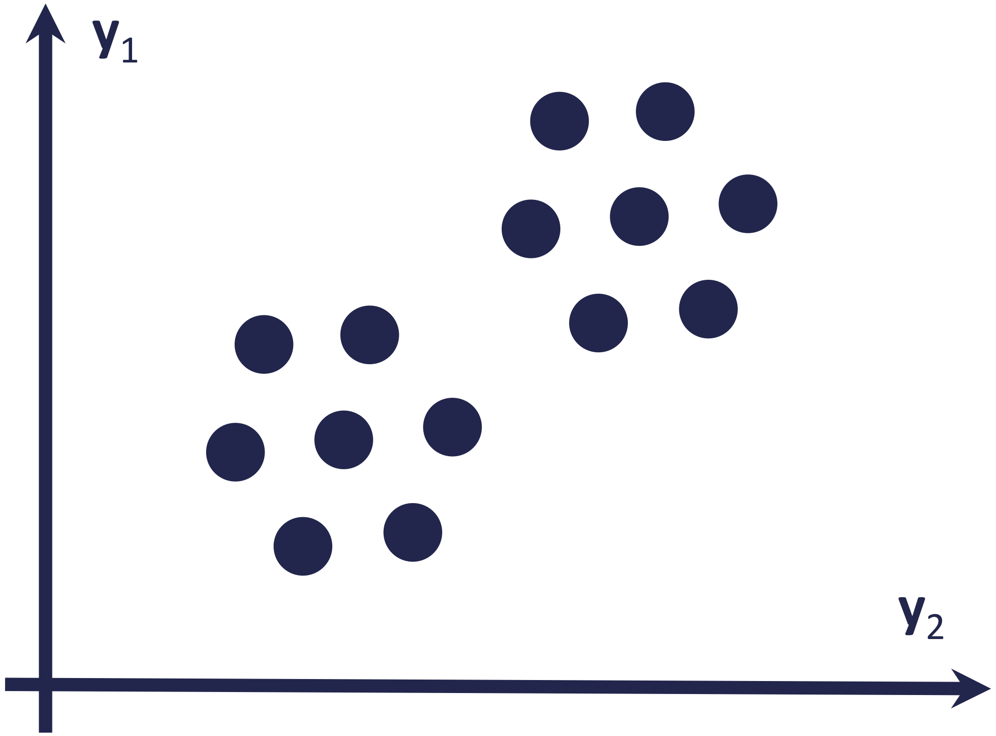
$$\{\mathbf{y}_1, \ldots, \mathbf{y}_d\}$$

are the first *d* eigenvectors of the Laplacian matrix

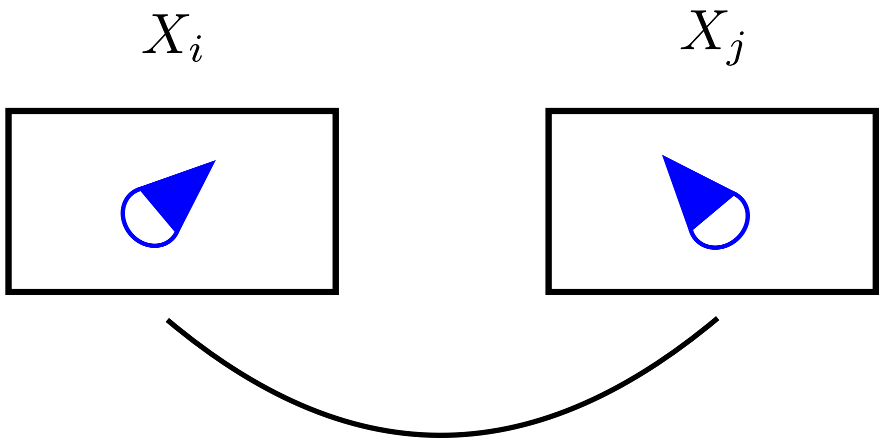
Summing up

Analyse the **distribution** of points

- ☐ $\{X_1, X_2, \ldots, X_K\} \quad X_i \in \mathbb{R}^{N \times T}$
- ☐ $\mathbf{D}_{ij} = d(X_i, X_j)$
- ☐ $\mathbf{W}_{ij} = \exp\left(-\frac{\mathbf{D}_{ij}^2}{\varepsilon}\right)$
- ☐ $\{Y_1, Y_2, \ldots, Y_K\} \quad Y_i \in \mathbb{R}^d$

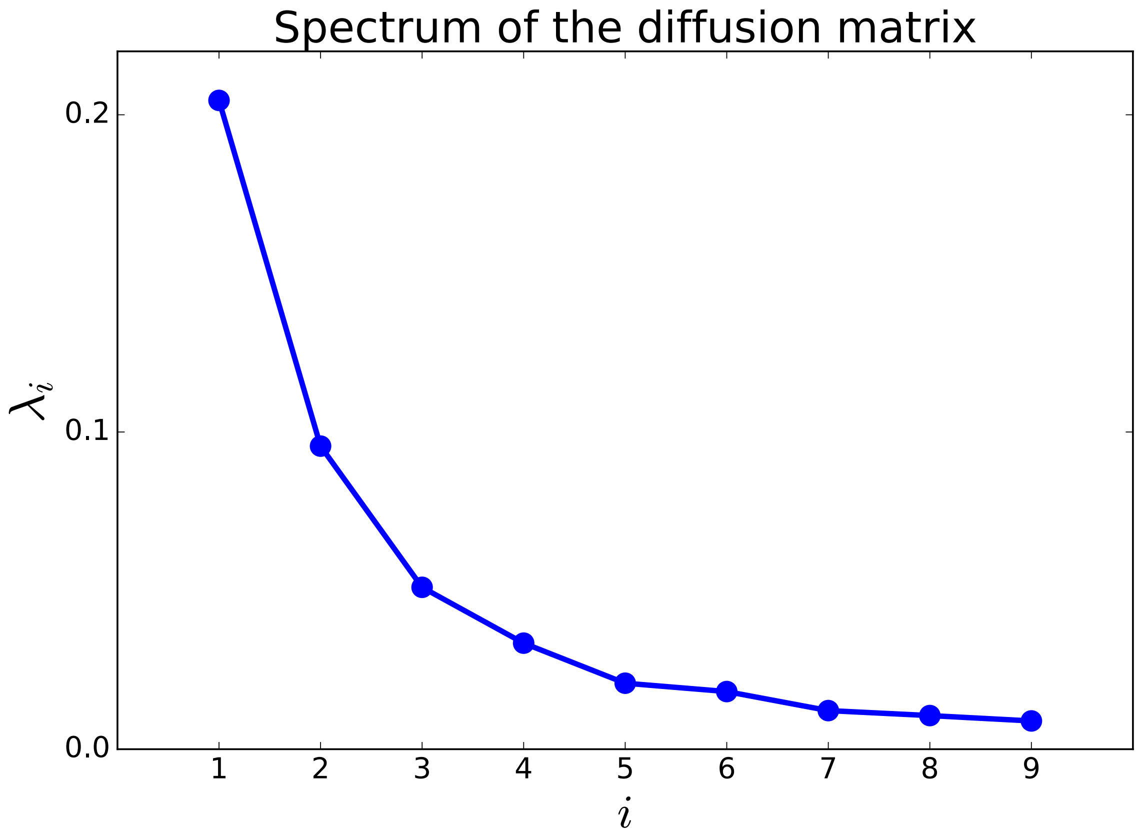


Example with **video** frames

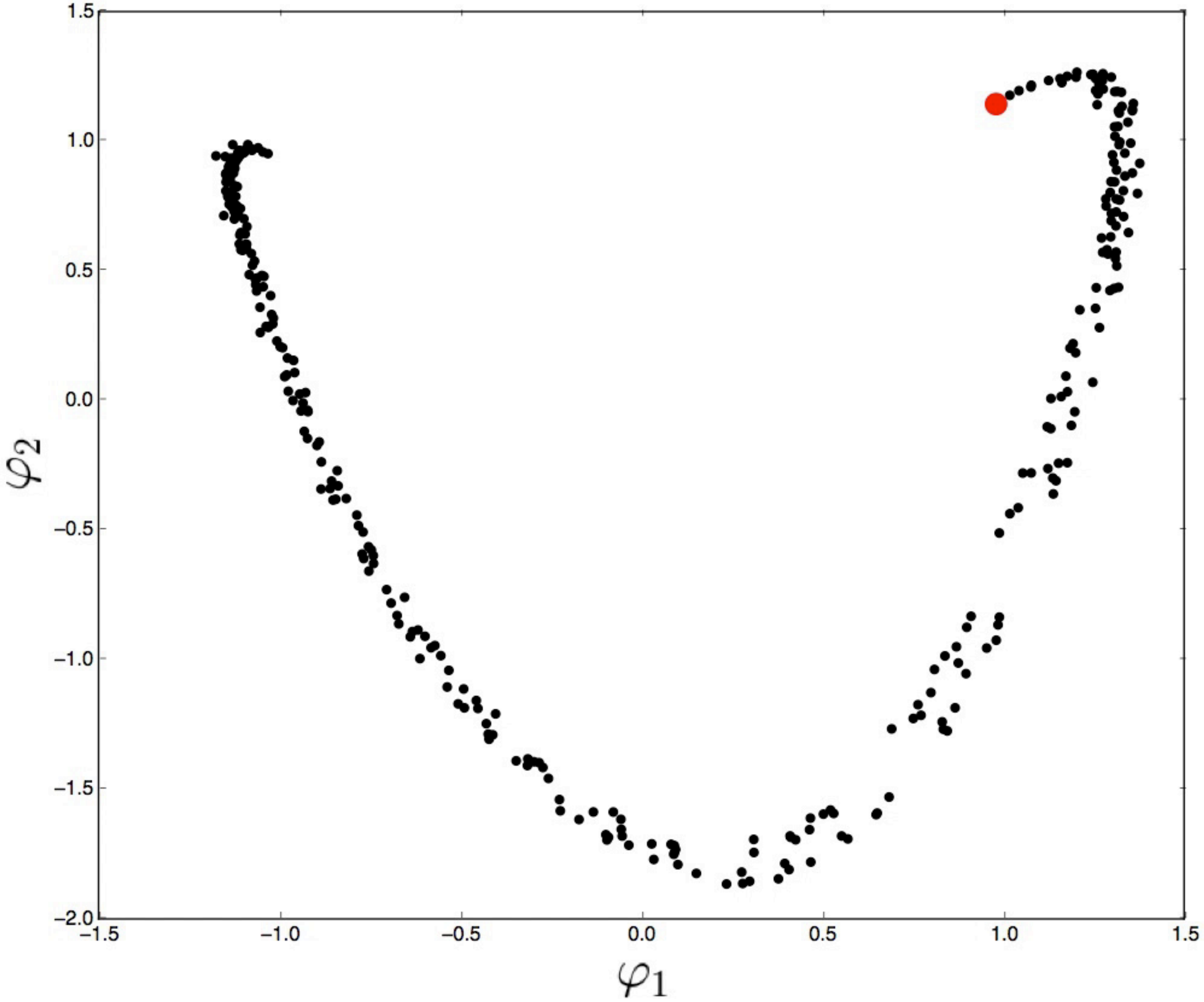


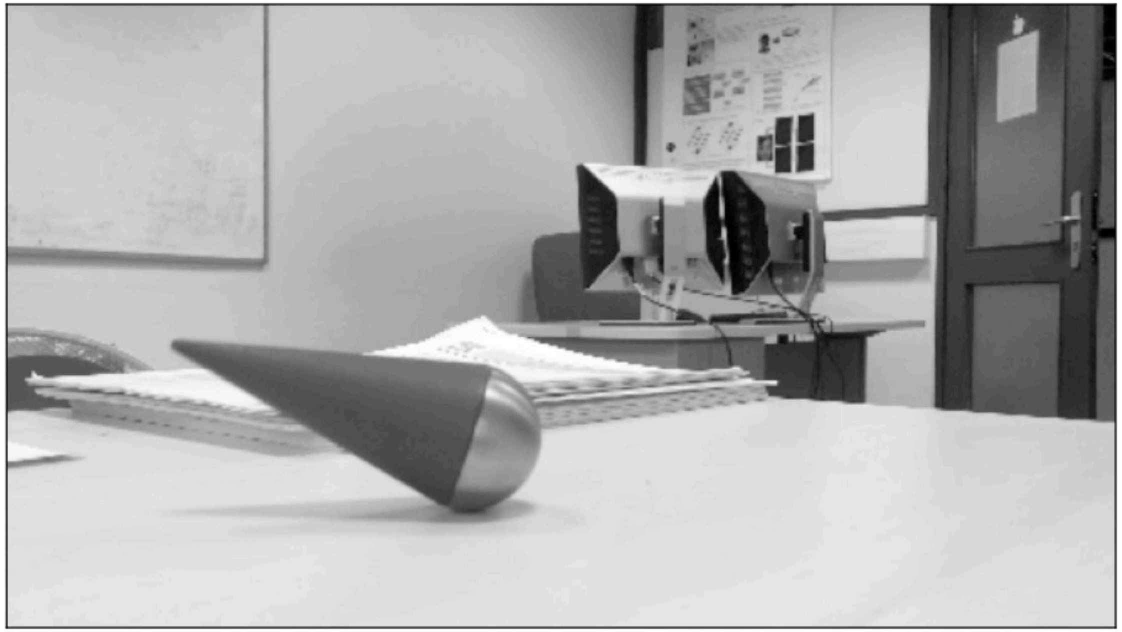
$$d^2(X_i, X_j) = \sum_k \sum_\ell \Big(X_i(k,\ell) - X_j(k,\ell) \Big)^2$$

(other distances could be used)



Diffusion Matrix = Laplacian Matrix


1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
φ_1
φ_2



How to define a distance between multivariate time-series ?

Distance between Multivariate Time-Series

For a zero-mean Gaussian **wide-sense stationary** multivariate time-series $\boldsymbol{x}(n)$ one has that :

(Autocovariances) $$R_{\boldsymbol{x}\boldsymbol{x}}(\tau) = \mathbb{E}\left[\boldsymbol{x}(n)\boldsymbol{x}(n-\tau)^T\right]$$

(Cross-spectral density) $$S_{\boldsymbol{x}\boldsymbol{x}}(f) = \sum_{k=-\infty}^{+\infty} R_{\boldsymbol{x}\boldsymbol{x}}(k)e^{-j2\pi fk}$$

describes all the **statistical features** of $\boldsymbol{x}(n)$

Distance between Multivariate Time-Series

For a zero-mean Gaussian **wide-sense stationary** multivariate time-series $\boldsymbol{x}(n)$ one has that :

(Autocovariances) $$R_{\boldsymbol{x}\boldsymbol{x}}(\tau) = \mathbb{E}\left[\boldsymbol{x}(n)\boldsymbol{x}(n-\tau)^T\right]$$

(Cross-spectral density) $$S_{\boldsymbol{x}\boldsymbol{x}}(f) = \sum_{k=-\infty}^{+\infty} R_{\boldsymbol{x}\boldsymbol{x}}(k)e^{-j2\pi fk}$$

describes all the **statistical features** of $\boldsymbol{x}(n)$

Two multivariate time series can be compared based on the **parameters** of their statistical descriptions.

- Compare covariance matrices
- Compare cross-spectral densities
- Compare MVAR models
- Etc.

Distance between Multivariate Time-Series

Distance based on the **covariance matrices** of two time-series :

$$d_C^2\big(\boldsymbol{x}_1(n), \boldsymbol{x}_2(n)\big) = \delta_R^2(C_1, C_2)$$

$$\delta_R^2(A, B) = \left\| \log\left(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}\right) \right\|_F^2$$

The C_i are **positive definite**

Affine-Invariant Riemannian distance

This distance can also be interpreted as the one obtained from the **Fisher-Rao** metric on the space of multivariate normal distributions

Distance between Multivariate Time-Series

Distance based on the **covariance matrices** of two time-series :

$$d_C^2\big(\boldsymbol{x}_1(n), \boldsymbol{x}_2(n)\big) = \delta_R^2(C_1, C_2)$$

$$\delta_R^2(A, B) = \left\| \log\left(A^{-\frac{1}{2}} B A^{-\frac{1}{2}}\right) \right\|_F^2$$

The C_i are **positive definite**

Affine-Invariant Riemannian distance

Distance based on the **cross-spectral density** matrices of two time-series :

$$d_S^2\big(\boldsymbol{x}_1(n), \boldsymbol{x}_2(n)\big) = \sum_{f_k \in \mathcal{F}} \delta_R^2\big(S_1(f_k), S_2(f_k)\big)$$ where $\mathcal{F}$ is a set of frequencies

Outline

Introduction

Manifold Learning for Time-Series Analysis

Some results

Concluding remarks

Results on **three** datasets with different kinds of EEG signals

BCI	SLEEP	EPILEPSY
BCI IV competition (public)	Prague's dataset (private)	Kaggle competition (public)

Results on **three** datasets with different kinds of EEG signals

BCI

BCI IV competition
(public)

SLEEP

Prague's dataset
(private)

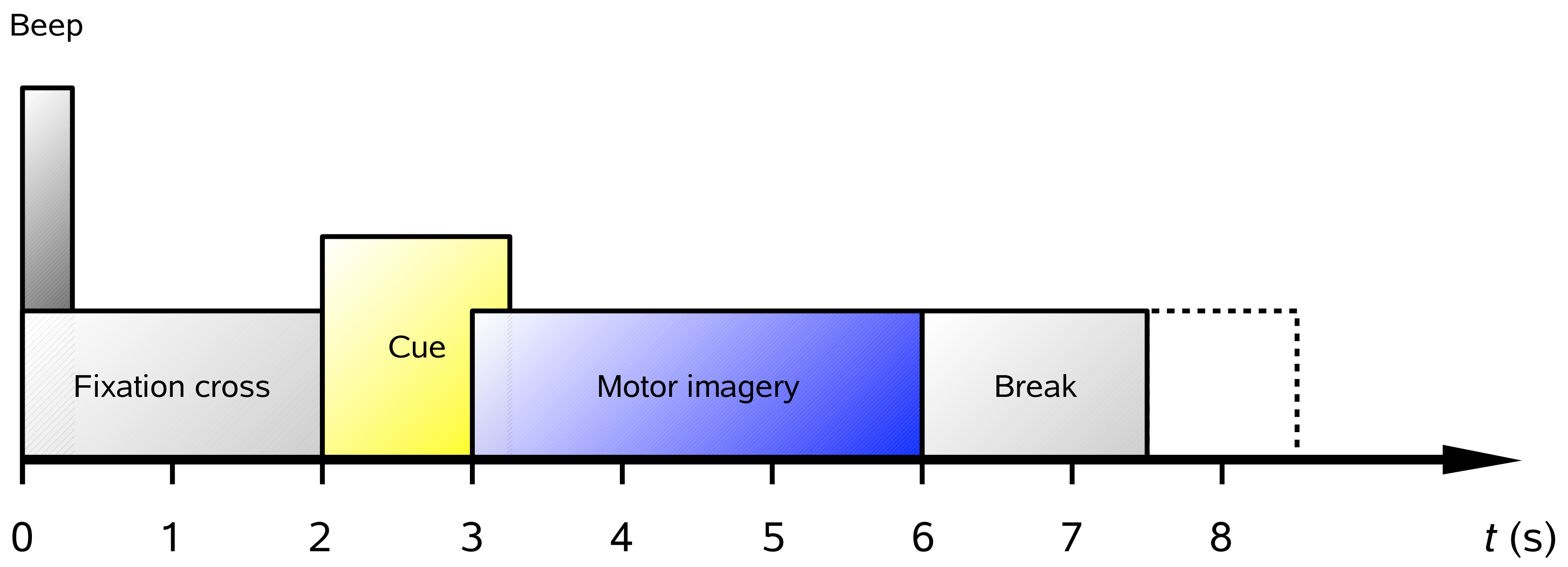
EPILEPSY

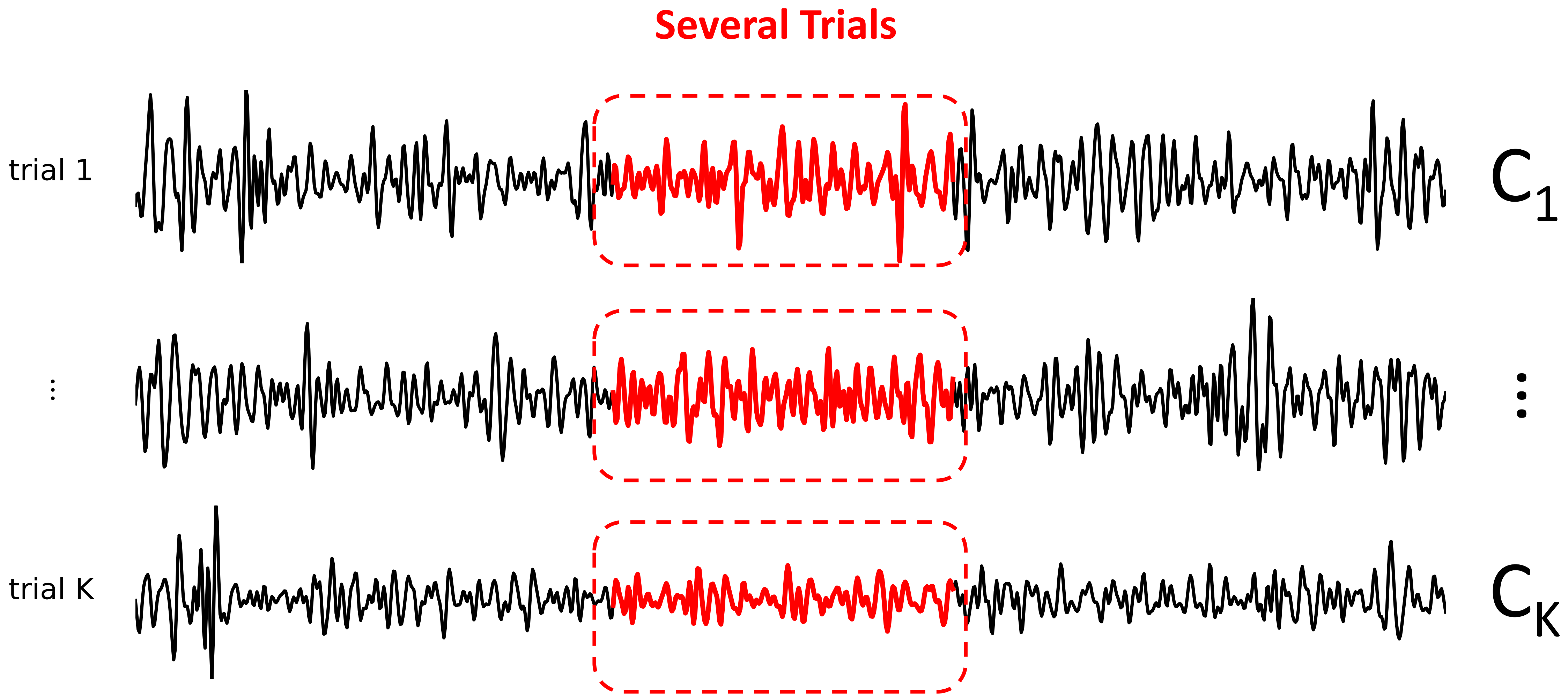
Kaggle competition
(public)

BCI dataset: BCI IV competition

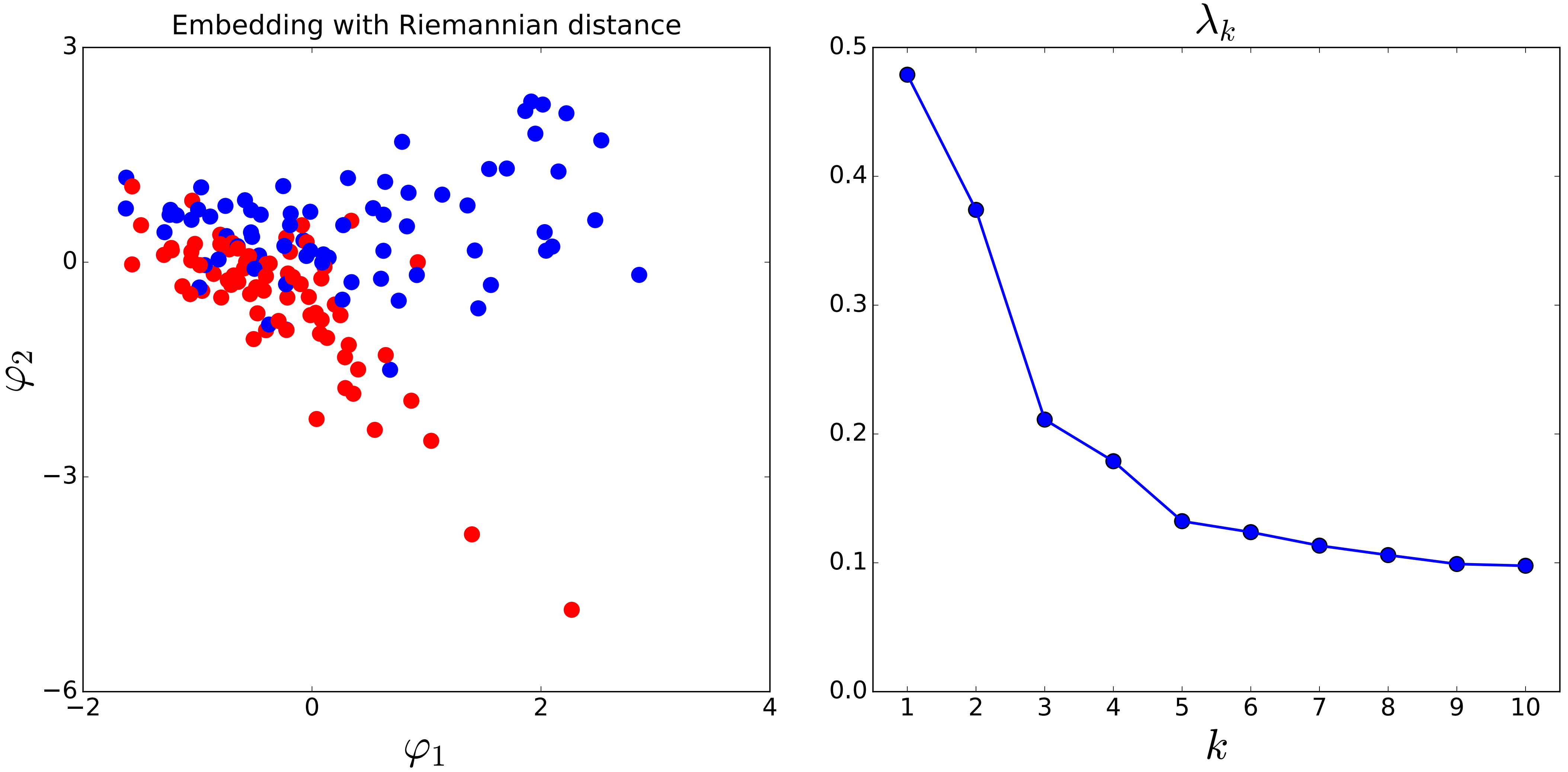
Public dataset with recordings from **motor imagery** BCI

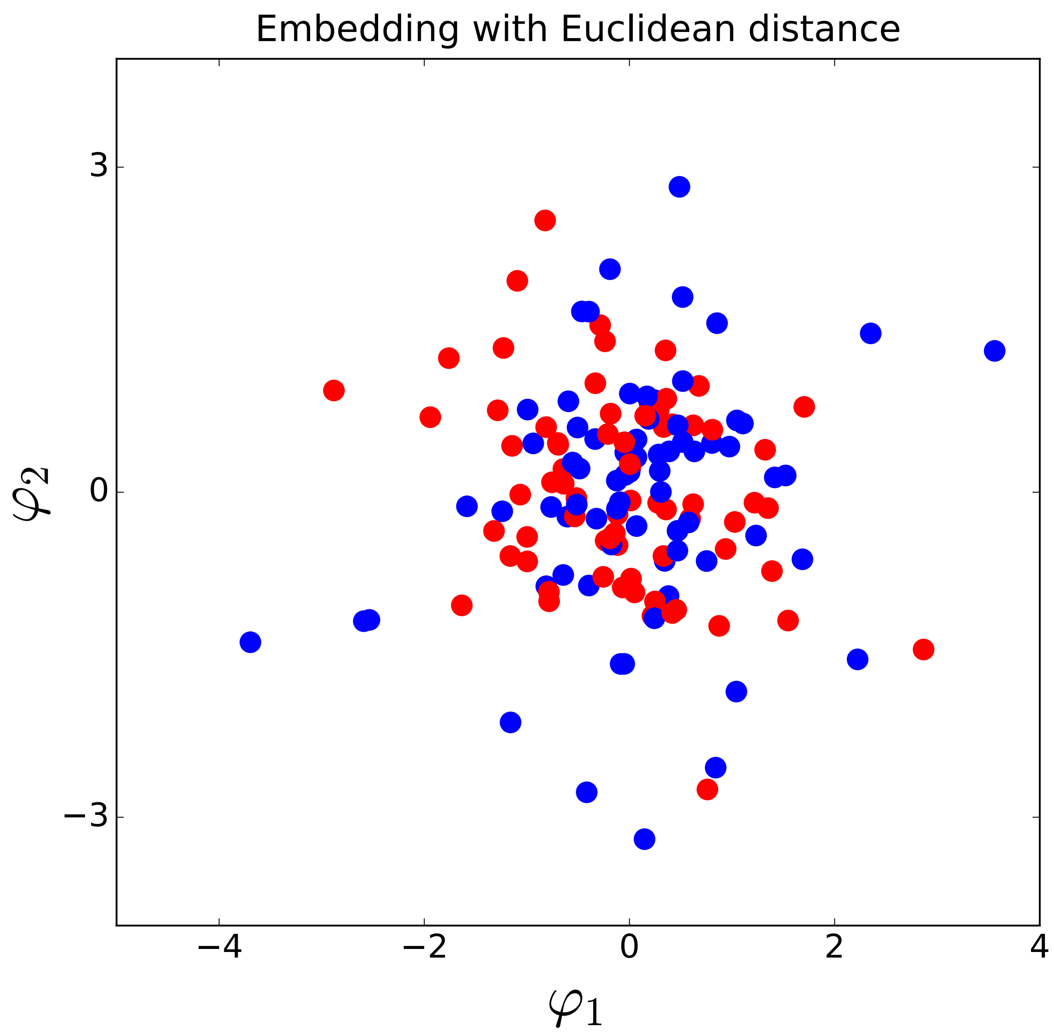
- EEG with 22 electrodes
- Four **classes** {left, right, tongue, feet} with 72 trials each
- Sampling frequency 250 Hz

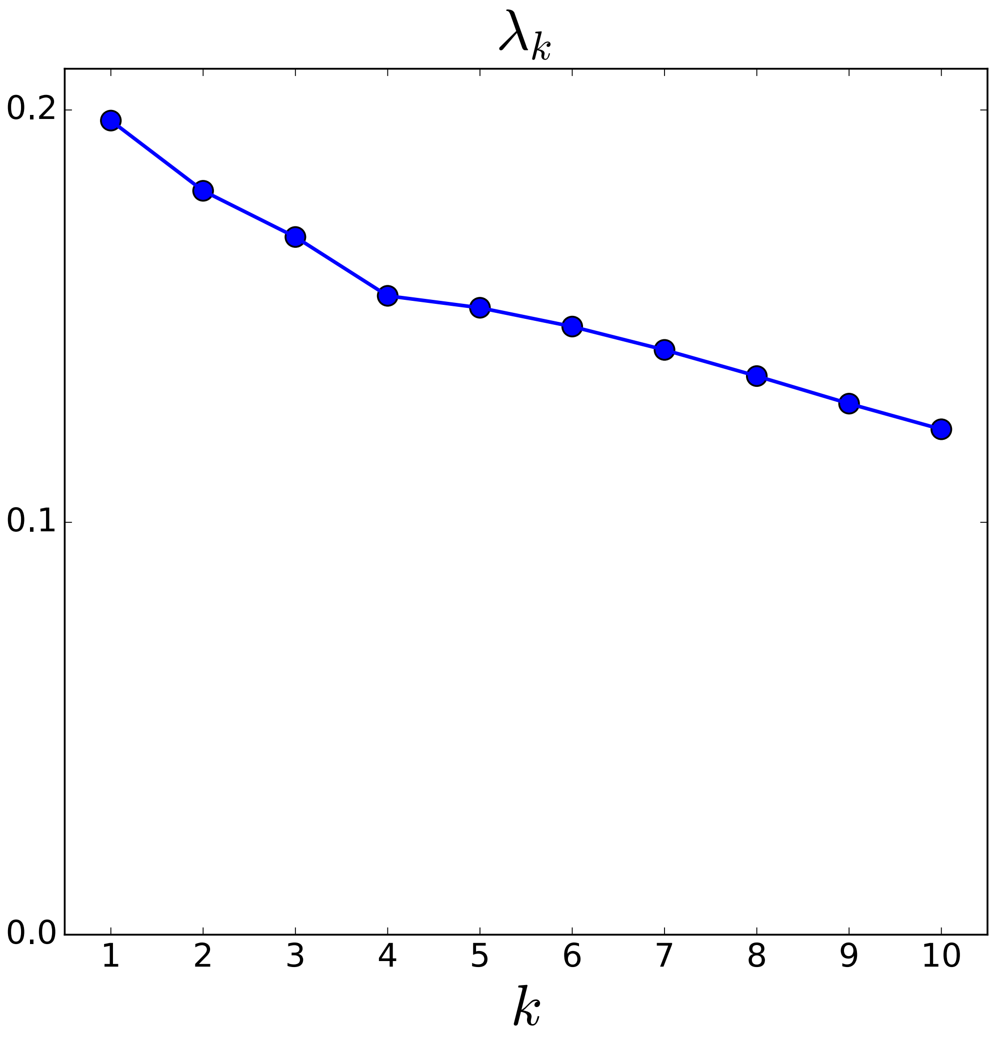


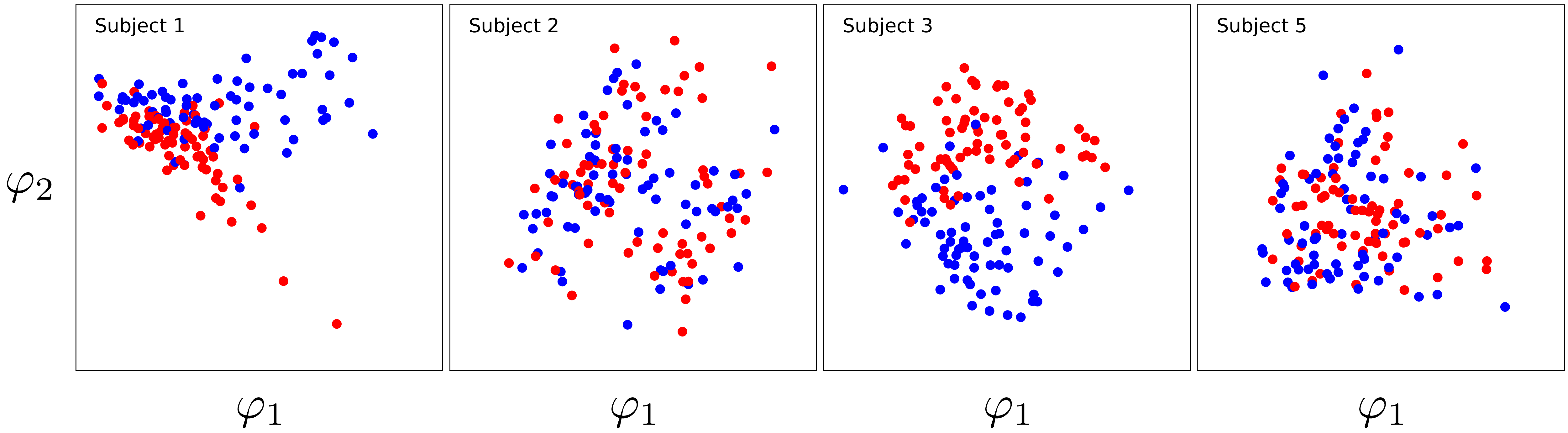

Several Trials
trial 1
⋮
trial K
C_1
⋮
C_K

gipsa-lab


Embedding with Riemannian distance
3
0
−3
−6
−2
0
2
4
φ_1
φ_2
λ_k
0.5
0.4
0.3
0.2
0.1
0.0
1
2
3
4
5
6
7
8
9
10
k

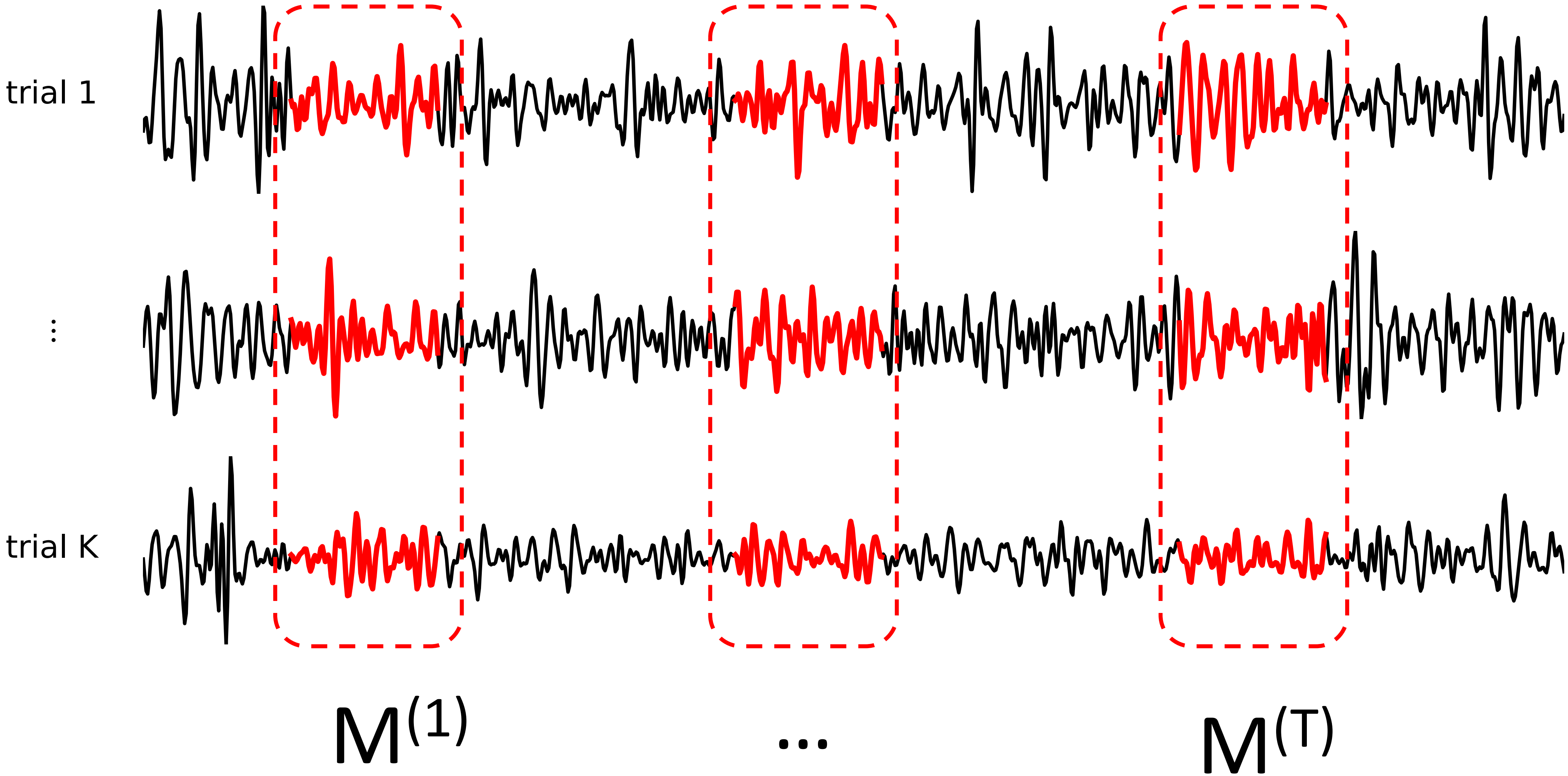

Embedding with Euclidean distance
3
0
−3
−4
−2
0
2
4
φ_2
φ_1


λ_k
0.2
0.1
0.0
1
2
3
4
5
6
7
8
9
10
k


Subject 1
Subject 2
Subject 3
Subject 5
φ_2
φ_1
φ_1
φ_1
φ_1

https://plcrodrigues.github.io/riemann-lab/

Sliding Window



Results on **three** datasets with different kinds of EEG signals

BCI

BCI IV competition
(public)

SLEEP

Prague's dataset
(private)

EPILEPSY

Kaggle competition
(public)

Sleep dataset: Prague's dataset

Private dataset from "National Institute for Mental Health" in Prague

- EEG with 19 electrodes, EOG with 2 electrodes, EMG with 2 electrodes
- Four **sleep stages** {S1, S2, S3, REM} with ~200 two-seconds clips for each

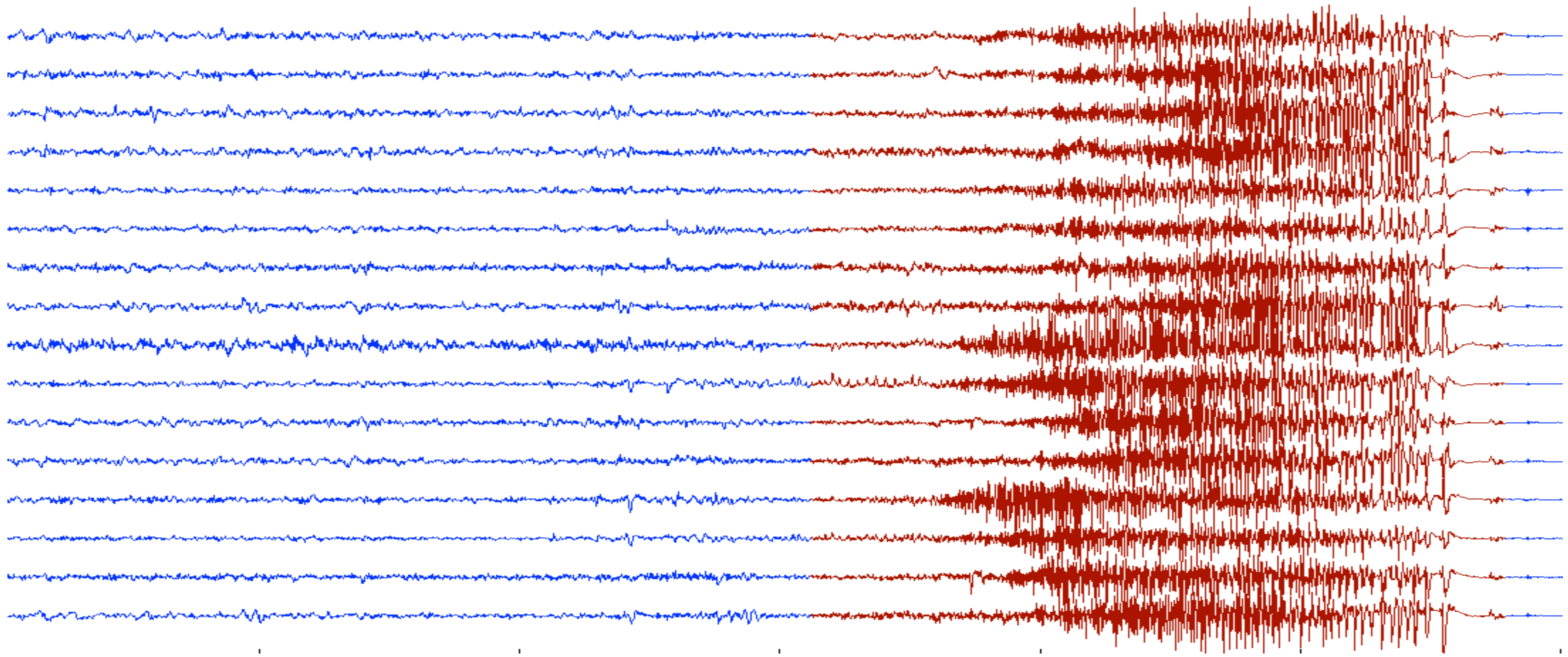
Results on **three** datasets with different kinds of EEG signals

BCI	SLEEP	EPILEPSY
BCI IV competition (public)	Prague's dataset (private)	Kaggle competition (public)

Epilepsy: Kaggle's competition dataset

Dataset used in a Kaggle competition for **epileptic** seizure detection

- **Intracranial** EEG with 36 electrodes sampled at 5000 Hz
- Dataset has several 1s clips with recordings from a few minutes **before** seizure onset (interictal)



Outline

Introduction

Manifold Learning for Time-Series Analysis

Some results

Concluding remarks

CONCLUSIONS

It’s important to use an **adequate** notion of distance between trials

Spectral embedding can be used for **static** and **dynamic** time-series analysis

PERSPECTIVES

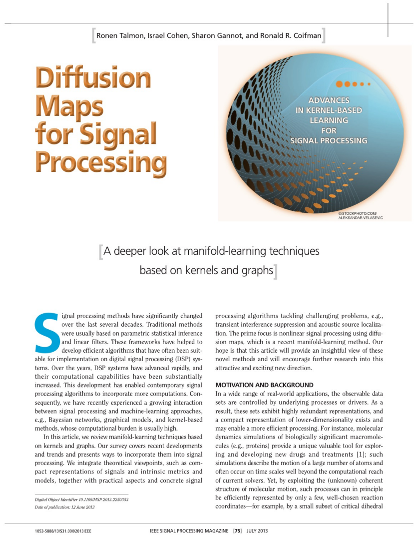
How to use the spectral embedding for **semi-supervised** classification ?

How to **compare** embeddings of two datasets ?

Can we detect **landmark** trials from the Laplacian matrix ?

Thank you **very much** for your attention :)

Questions ?



[Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald R. Coifman]

Diffusion Maps for Signal Processing

ADVANCES IN KERNEL-BASED LEARNING FOR SIGNAL PROCESSING

[A deeper look at manifold-learning techniques based on kernels and graphs]

Signal processing methods have significantly changed over the last several decades. Traditional methods were usually based on parametric statistical inference and linear filters. These frameworks have helped to develop efficient algorithms that have often been suitable for implementation on digital signal processing (DSP) systems. Over the years, DSP systems have advanced rapidly, and their computational capabilities have been substantially increased. This development has enabled contemporary signal processing algorithms to incorporate more computations. Consequently, we have recently experienced a growing interaction between signal processing and machine-learning approaches, e.g., Bayesian networks, graphical models, and kernel-based methods, whose computational burden is usually high.

In this article, we review manifold-learning techniques based on kernels and graphs. Our survey covers recent developments and trends and presents ways to incorporate them into signal processing. We integrate theoretical viewpoints, such as compact representations of signals and intrinsic metrics and models, together with practical aspects and concrete signal processing algorithms tackling challenging problems, e.g., transient interference suppression and acoustic source localization. The prime focus is nonlinear signal processing using diffusion maps, which is a recent manifold-learning method. Our hope is that this article will provide an insightful view of these novel methods and will encourage further research into this attractive and exciting new direction.

Digital Object Identifier 10.1109/MSP.2013.2250353
Date of publication: 12 June 2013

MOTIVATION AND BACKGROUND

In a wide range of real-world applications, the observable data sets are controlled by underlying processes or drivers. As a result, these sets exhibit highly redundant representations, and a compact representation of lower-dimensionality exists and may enable a more efficient processing. For instance, molecular dynamics simulations of biologically significant macromolecules (e.g., proteins) provide a unique valuable tool for exploring and developing new drugs and treatments [1]; such simulations describe the motion of a large number of atoms and often occur on time scales well beyond the computational reach of current solvers. Yet, by exploiting the (unknown) coherent structure of molecular motion, such processes can in principle be efficiently represented by only a few, well-chosen reaction coordinates—for example, by a small subset of critical dihedral

1053-5888/13/$31.00©2013IEEE IEEE SIGNAL PROCESSING MAGAZINE [75] JULY 2013

Ronen Talmon, Israel Cohen, Sharon Gannot, and Ronald Coifman (2013)
Diffusion maps for signal processing: A deeper look at manifold-learning techniques based on kernels and graphs

Stéphane Lafon (Ph.D. thesis, 2004)
Diffusion maps and Geometric Harmonics

Ronald Coifman and Stéphane Lafon (2006)
Diffusion maps

An Introduction to Manifold Learning

and its use in time series analysis

Pedro L. C. Rodrigues – Machine Learning Seekers of Truth – 6th December 2017