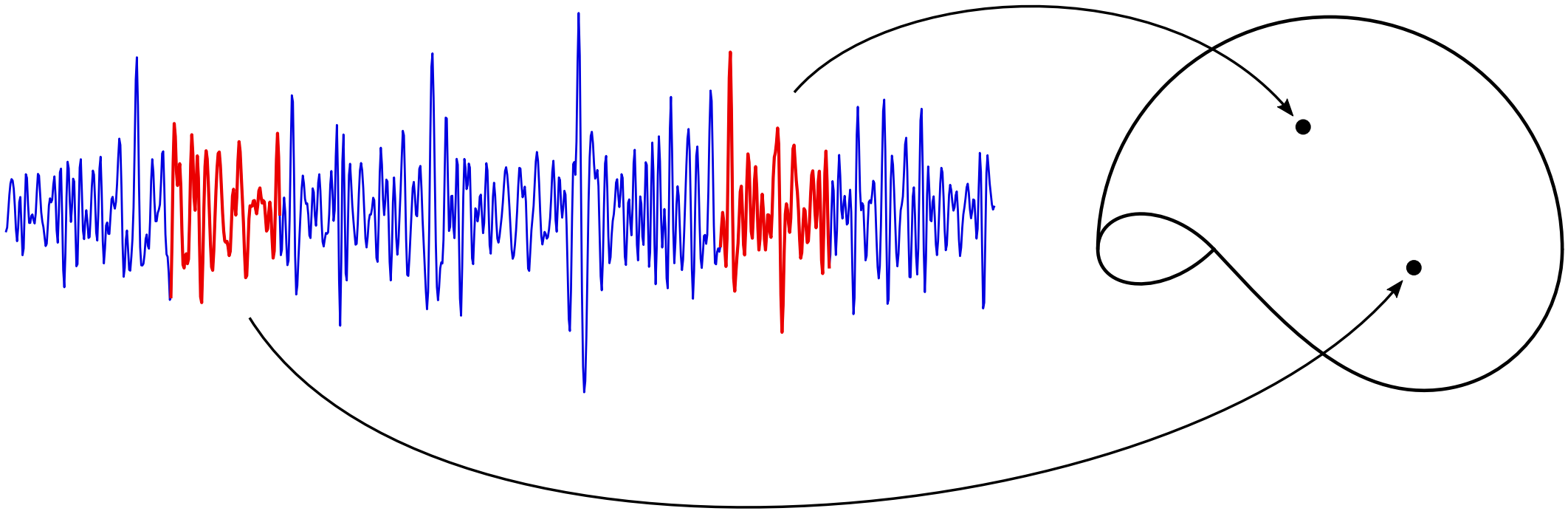
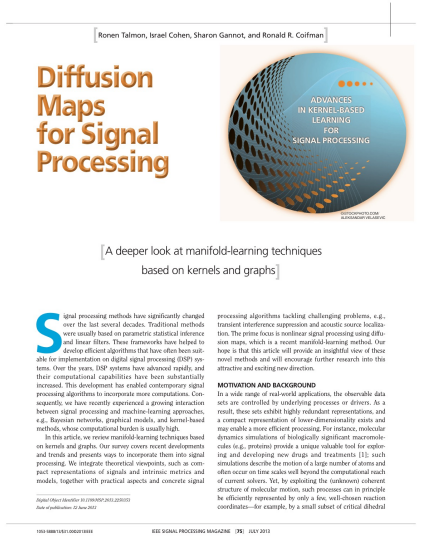
An introduction to diffusion maps

and their use in time series analysis

Pedro Rodrigues – VIBS team meeting, 17th March 2017





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.

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: 25 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

Diffusion maps

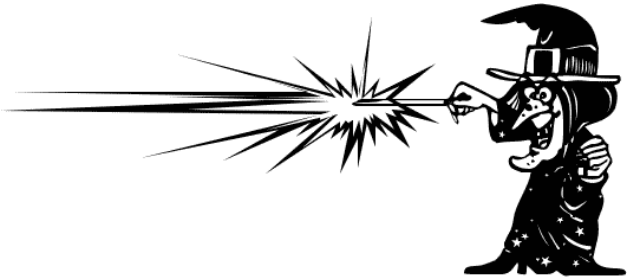
Time series analysis

Concluding remarks

Introduction

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

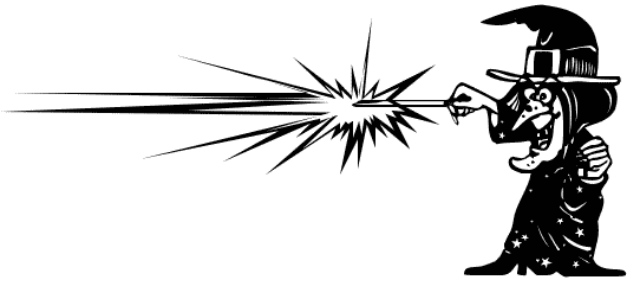
- Norms in $\mathbf{R}^n$ are not **numerically** equivalent when n is large
- Density estimation is **difficult** in big dimensions (needs too much data)
- Fast algorithms in low dimensions may become **prohibitively slow** in high dimensions



curse of dimensionality

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 too much data)
- Fast algorithms in low dimensions may become **prohibitively slow** in high dimensions

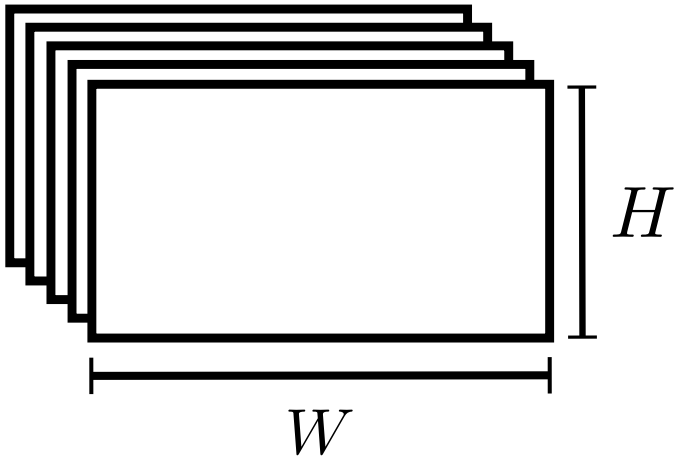


curse of dimensionality

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

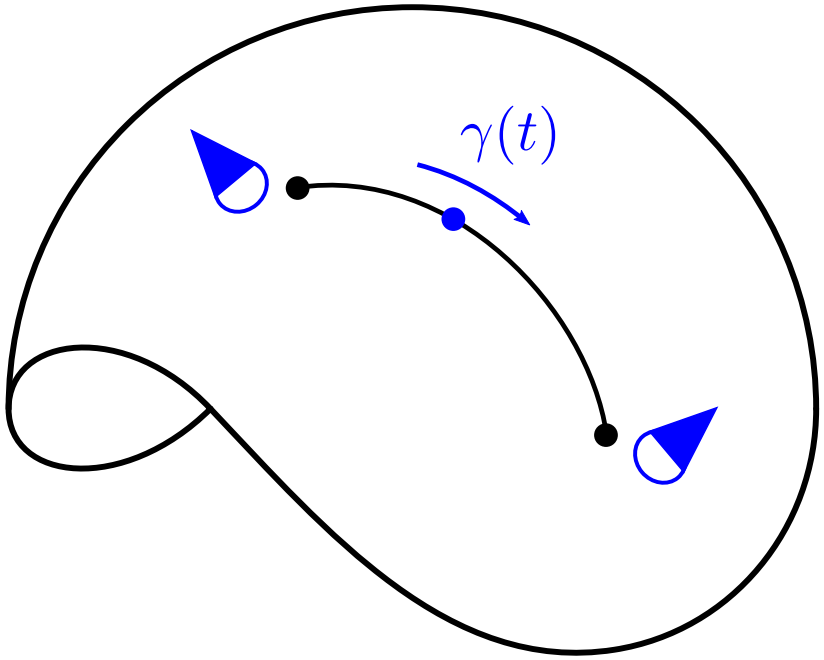
Fundamental assumption: data lies on (or close to) a low-dimensional manifold

Dimension reduction techniques for visualization, feature selection, compression, etc.



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

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

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

PCA - Principal component analysis

MDS - Multi-dimensional scaling

LLE - Local-linear embedding

- Classical techniques like PCA and MDS for **linear** dimension reduction. Kernel PCA for **nonlinear**.
- Two papers in Science magazine (2000) presenting **manifold learning** techniques: Isomap and LLE
- Previous methods are all **global** (except LLE)
- Laplacian eigenmaps (2003) and Diffusion maps (2004) search for an embedding that respects **local** geometry

"Think globally, act locally"

Diffusion maps

The algorithm consists of **three** steps

1) Build a **graph** with samples as vertices and **weights** defined using a kernel

2) Define a **Markov process** on the graph (normalize **W** matrix)

3) Obtain a **nonlinear mapping** of the samples into a new embedded space

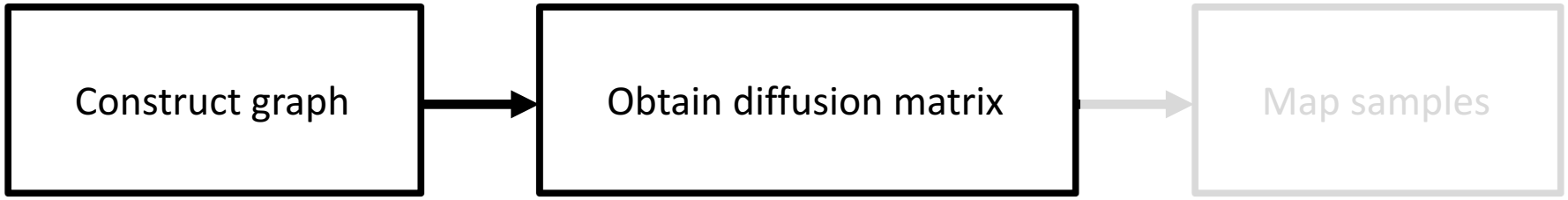
"A kernel specifies the **local geometry** of the data and captures some geometric features of interest. The Markov chain describes how this local information **propagates** along the graph"

Define a graph Ω whose weights represent **pairwise affinity** between samples

$$\mathbf{W}(i,j) = \exp\left(-\frac{d(X_i, X_j)^2}{2\epsilon^2}\right)$$

ϵ scale parameter imposes a notion of **locality**

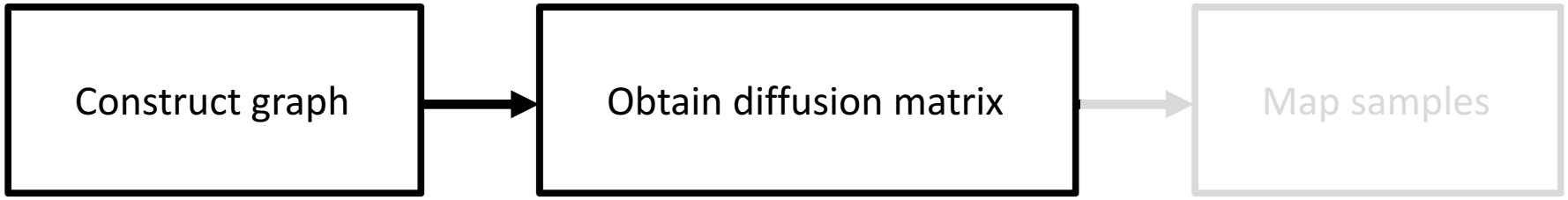
$d(X_i, X_j)$ measure of **similarity** between samples (application-specific)



Normalize the rows of matrix **W** so to have a probability transition (diffusion) matrix

$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \qquad \mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$$

$$\mathbf{P}(i,j) \geq 0 \qquad \sum_j \mathbf{P}(i,j) = 1$$



Normalize the rows of matrix **W** so to have a probability transition (diffusion) matrix

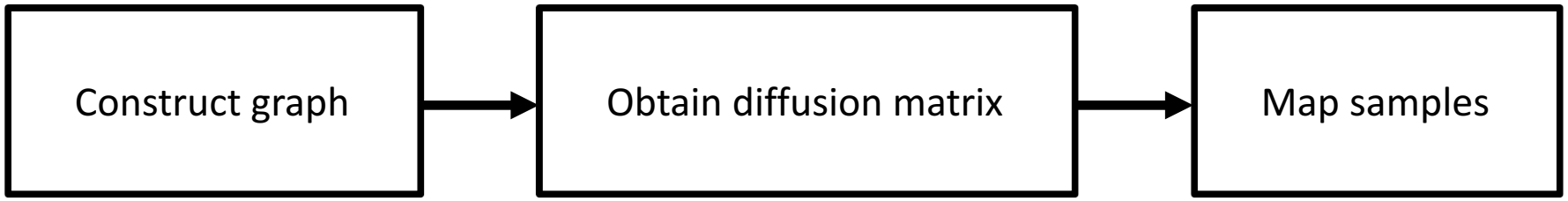
$$\mathbf{P} = \mathbf{D}^{-1}\mathbf{W} \qquad \mathbf{D}(i,i) = \sum_j \mathbf{W}(i,j)$$

probability of going from *i* to *j* in a **random walk** over the graph (diffusion process)

$$\mathbf{P}(i,j) = \sum_{k \geq 0} \lambda_k \boldsymbol{\varphi}_k(i) \boldsymbol{\psi}_k(j)$$

$\{\boldsymbol{\psi}_k\}$ **left** eigenvectors

$\{\boldsymbol{\varphi}_k\}$ **right** eigenvectors



Check the **decay** of the eigenvalues and choose a dimension *d*

Embed the data points as **coordinates** of the right eigenvectors of **P**

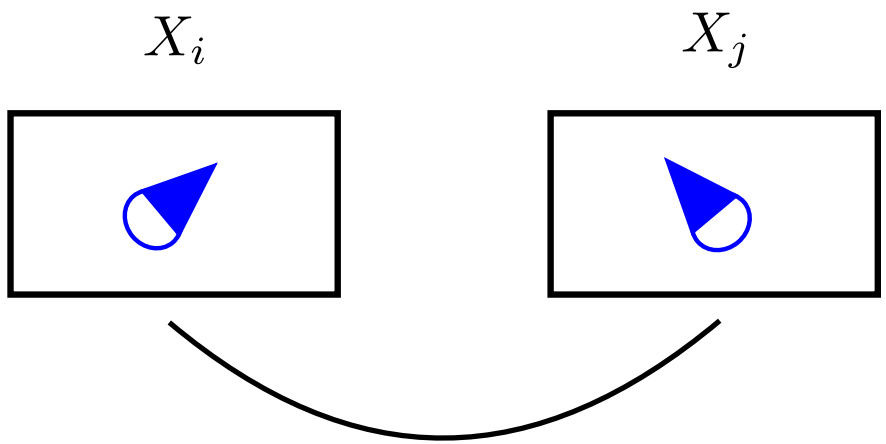
$$X_i \in \mathbb{R}^n \Rightarrow \Phi(X_i) = \begin{bmatrix} \lambda_1 \boldsymbol{\varphi}_1(i) \\ \vdots \\ \lambda_d \boldsymbol{\varphi}_d(i) \end{bmatrix} \in \mathbb{R}^d$$

$$\mathbf{P}\boldsymbol{\varphi}_k = \lambda_k \boldsymbol{\varphi}_k$$

$$\boldsymbol{\varphi}_k \in \mathbb{R}^N$$

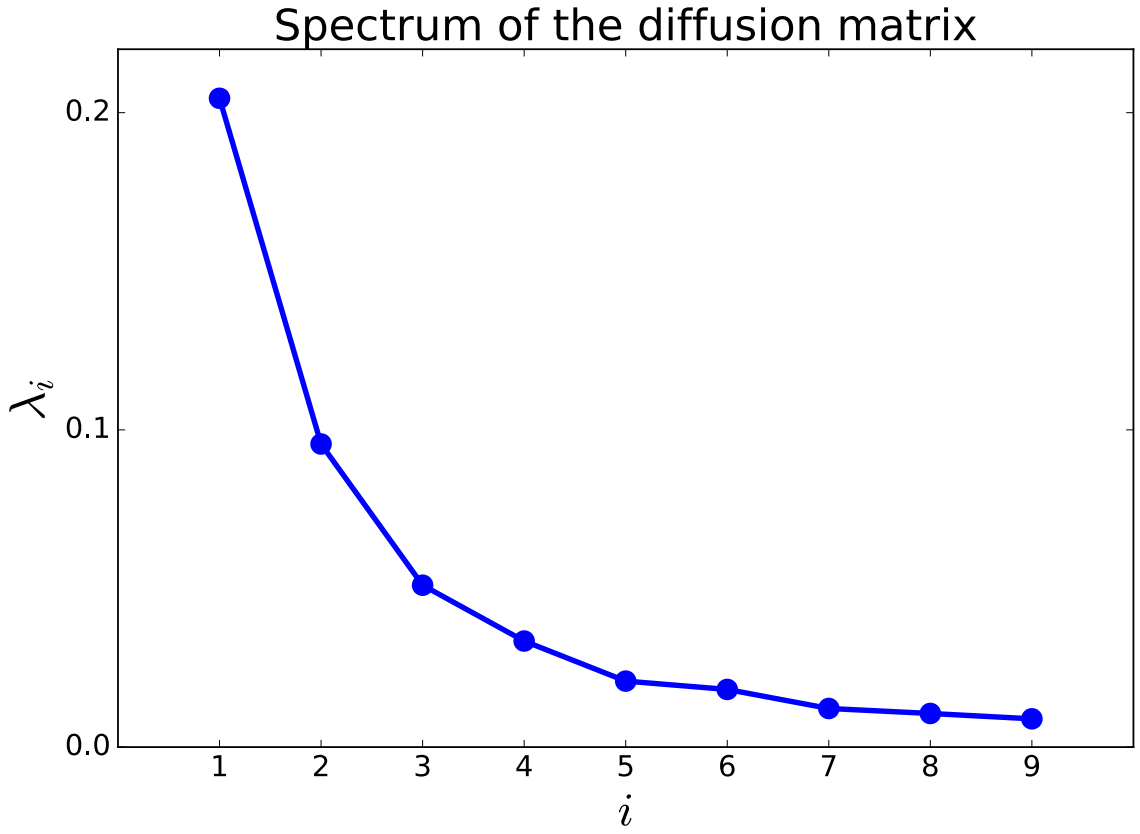
N is the number of samples

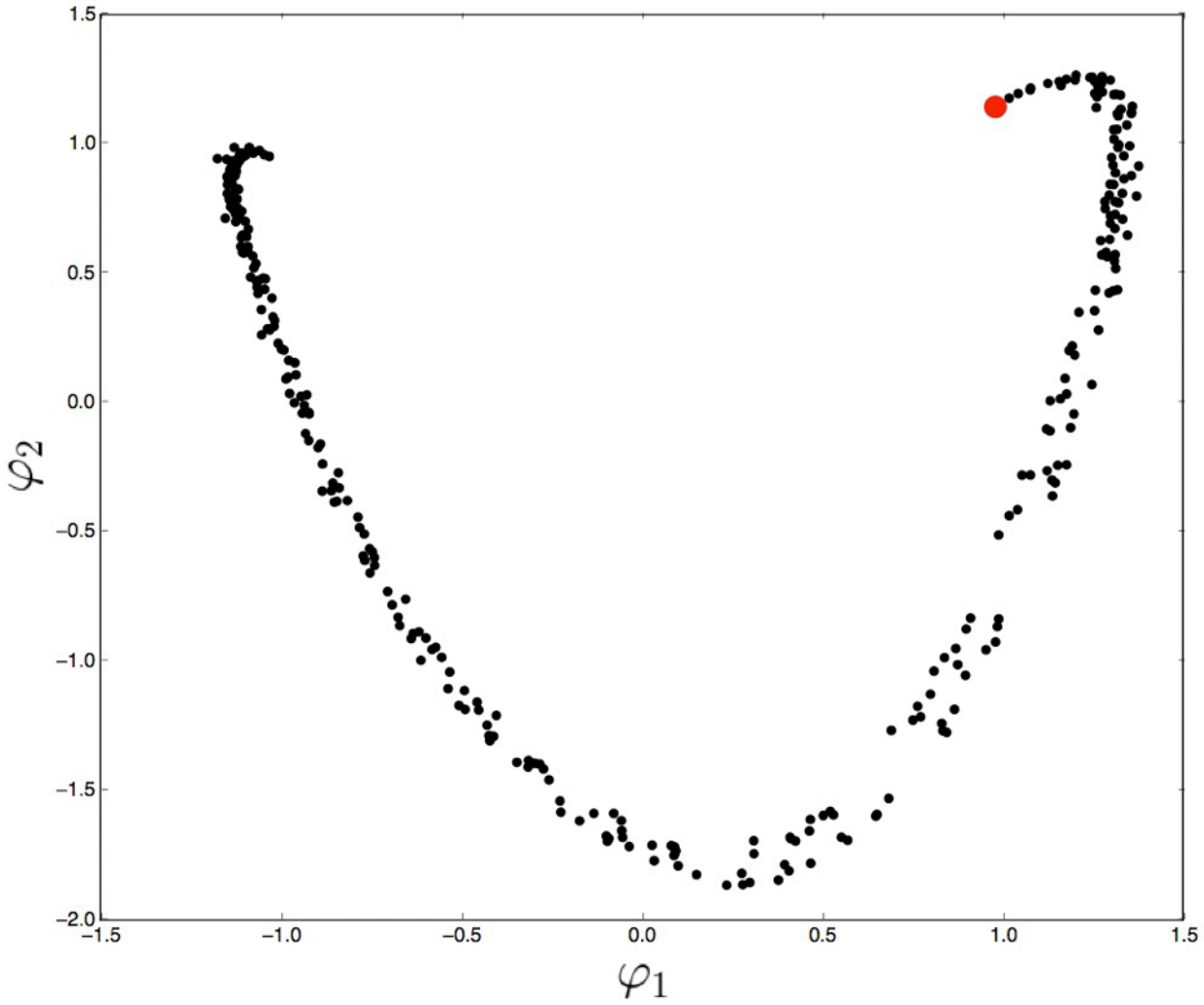
Example with **video** frames

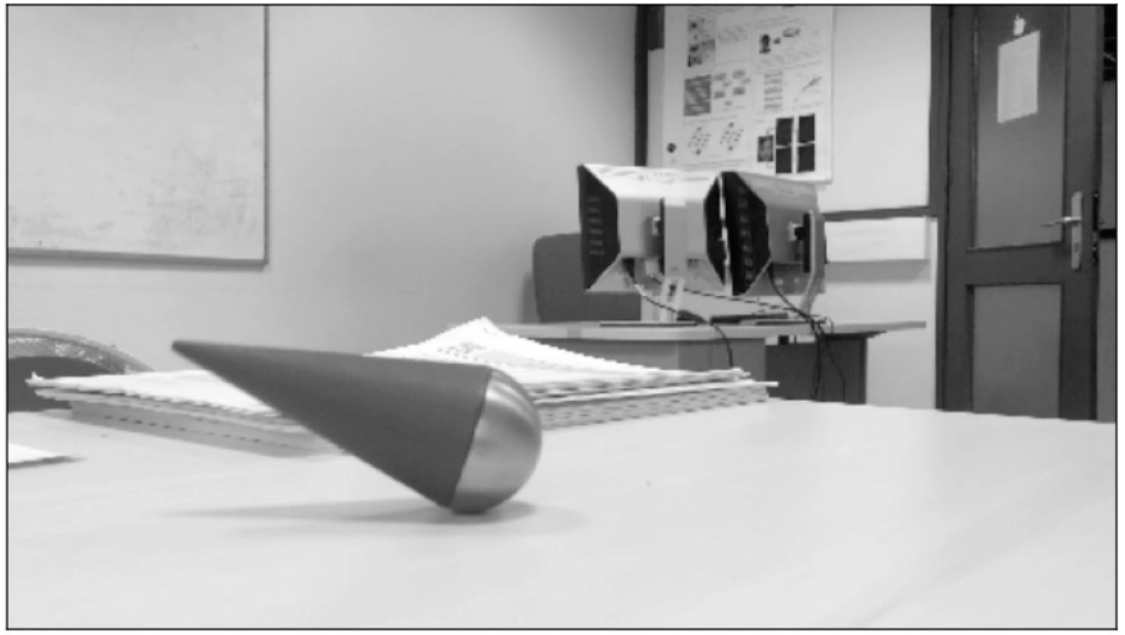


$$d^2(X_i, X_j) = \sum_k \sum_\ell \Big(X_i(k,\ell) - X_j(k,\ell) \Big)^2$$

(other distances could be used)



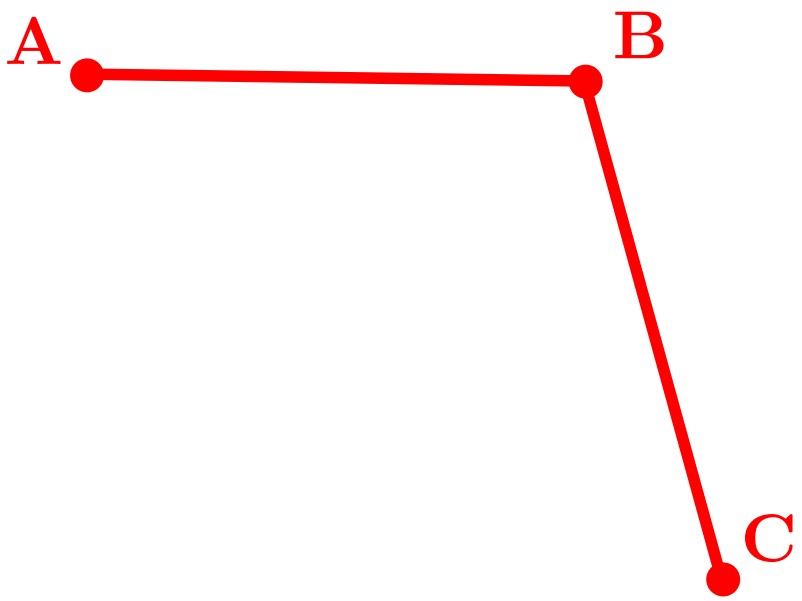

1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
φ_2
φ_1



The embedding gives good results, but **why**?

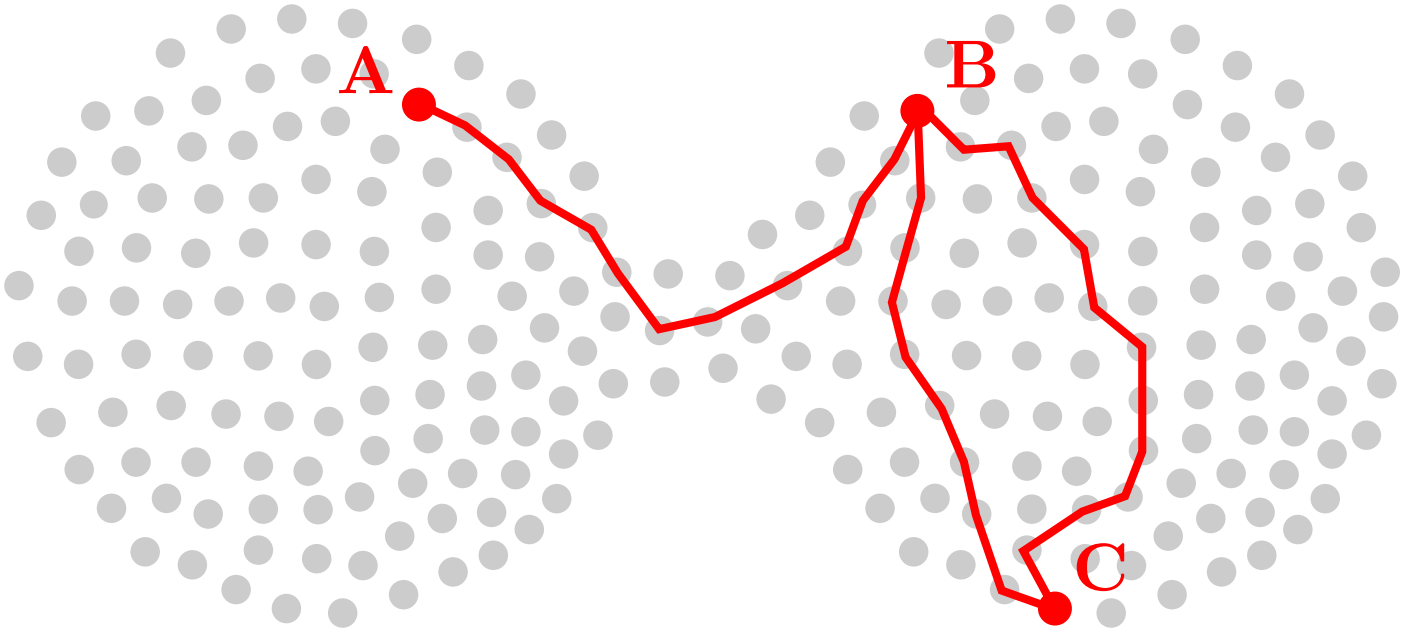
The embedding gives good results, but **why**?

It uses the notion of **diffusion distance**, a measure of connectivity between points in a dataset



The embedding gives good results, but **why**?

It uses the notion of **diffusion distance**, a measure of connectivity between points in a dataset



The embedding gives good results, but **why**?

It uses the notion of **diffusion distance**, a measure of connectivity between points in a dataset

$$D^2(X_i, X_j) = \sum_{y \in \Omega} \frac{1}{\pi(y)} \Big(\mathbf{P}(i, y) - \mathbf{P}(j, y) \Big)^2$$

which is the weighted L_2 distance between conditional distributions $\mathbf{P}(i, \cdot)$ and $\mathbf{P}(j, \cdot)$

$\pi(y)$ is the **stationary** distribution of the Markov chain defined on the graph Ω (first left eigenvector of **P**)

The embedding gives good results, but **why**?

It uses the notion of **diffusion distance**, a measure of connectivity between points in a dataset

$$D^2(X_i, X_j) = \sum_{y \in \Omega} \frac{1}{\pi(y)} \Big(\mathbf{P}(i, y) - \mathbf{P}(j, y) \Big)^2$$

1) Points are closer if they are highly **connected** in the graph (clusters)
2) Distances between two points depends on **all possible paths** between them (robust)
3) Distances between two points take into account **all evidences** relating them (good for inference)

Using the decomposition of the diffusion matrix

probability of going from *i* to *j* in a **random walk** over the graph (diffusion process)

$$\mathbf{P}(i,j) = \sum_{k \geq 0} \lambda_k \boldsymbol{\varphi}_k(i) \boldsymbol{\psi}_k(j)$$

$\{\boldsymbol{\psi}_k\}$ **left** eigenvectors

$\{\boldsymbol{\varphi}_k\}$ **right** eigenvectors

we can show that

$$D^2(X_i, X_j) = \sum_{k \geq 0} \lambda_k^2 \Big(\boldsymbol{\varphi}_k(i) - \boldsymbol{\varphi}_k(j) \Big)^2$$

and so

$$D^2(X_i, X_j) \simeq \sum_{k=1}^{d} \lambda_k^2 \Big(\boldsymbol{\varphi}_k(i) - \boldsymbol{\varphi}_k(j) \Big)^2$$

$$\boldsymbol{\varphi}_0 = \begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix}$$

$$D^2(X_i, X_j) \simeq \sum_{k=1}^{d} \lambda_k^2 \Big(\boldsymbol{\varphi}_k(i) - \boldsymbol{\varphi}_k(j) \Big)^2$$

this approximation can be interpreted as the **Euclidean** distance between two vectors

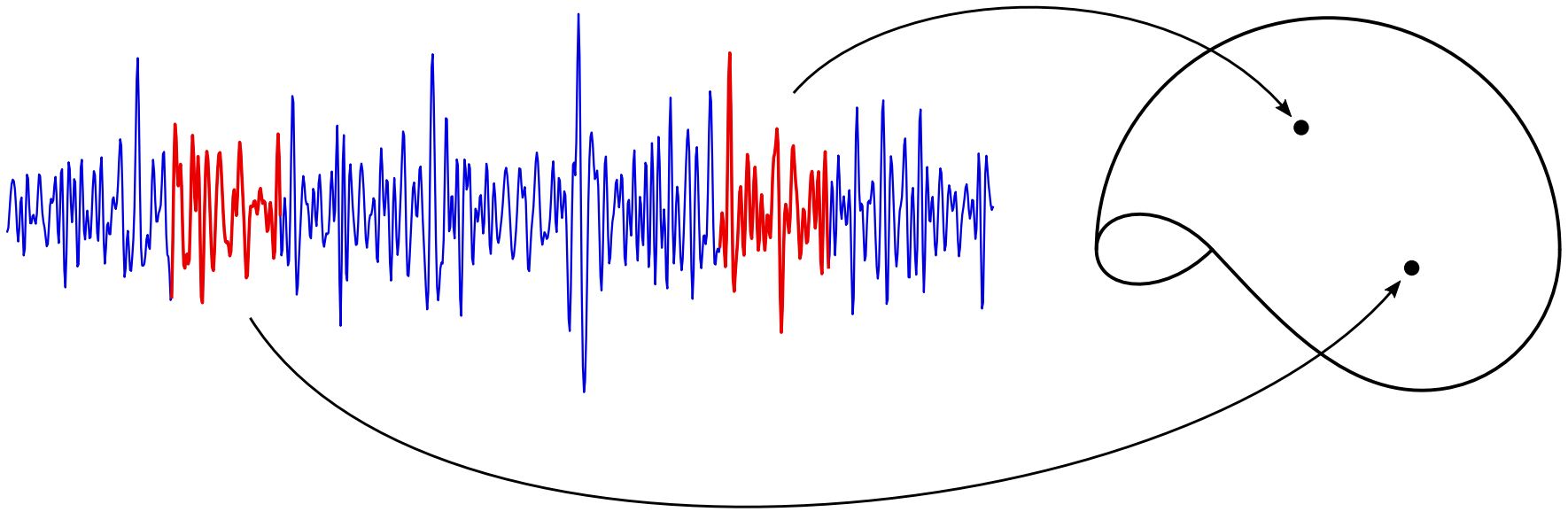
$$D^2(X_i, X_j) \simeq \left\| \Phi(X_i) - \Phi(X_j) \right\|^2$$

$$\Phi(X_i) = \begin{bmatrix} \lambda_1 \boldsymbol{\varphi}_1(i) \\ \vdots \\ \lambda_d \boldsymbol{\varphi}_d(i) \end{bmatrix} \qquad \Phi(X_j) = \begin{bmatrix} \lambda_1 \boldsymbol{\varphi}_1(j) \\ \vdots \\ \lambda_d \boldsymbol{\varphi}_d(j) \end{bmatrix}$$

Time series analysis

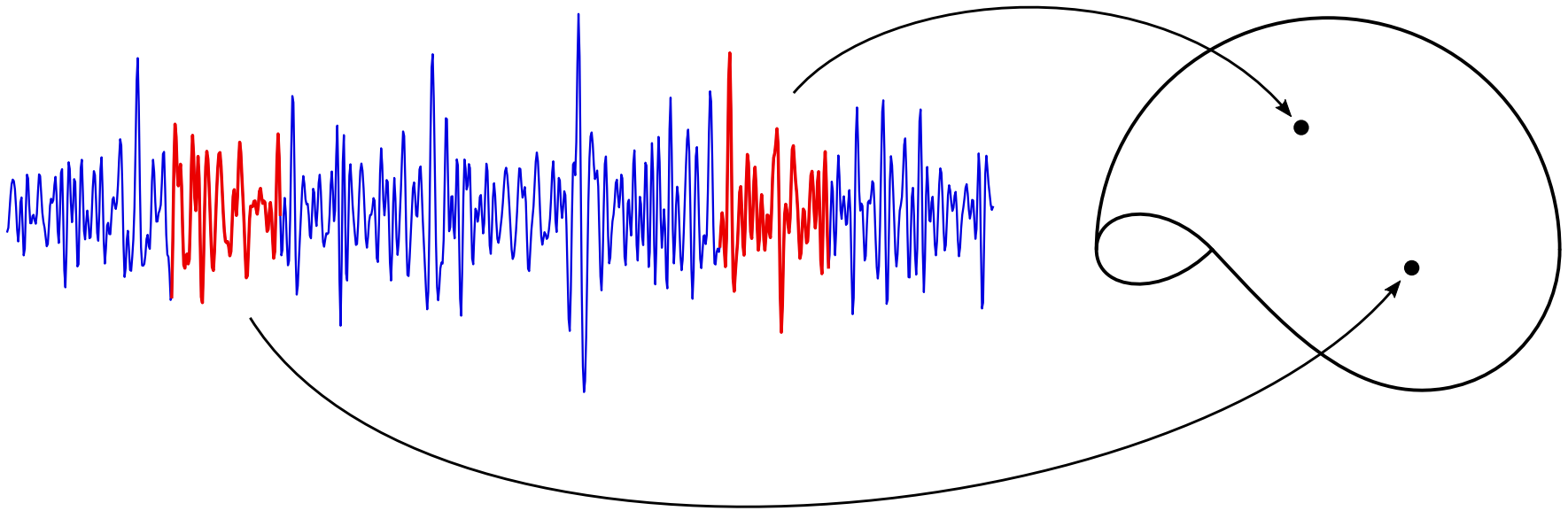
Two **bottlenecks** for time series analysis:

- Measurements are **realizations** of a stochastic process
- **Sequential** implementation



Two **bottlenecks** for time series analysis:

- Measurements are **realizations** of a stochastic process
- **Sequential** implementation



Describe signals in **state-space** and define an intrinsic affinity metric

$$\boldsymbol{x}(t) = f(\boldsymbol{\theta}(t))$$

$\boldsymbol{x}(t)$: measurement

$\boldsymbol{\theta}(t)$: state

f: map from state-space to measurement space (possibly nonlinear and/or probabilistic)

Use an **observer operator** to describe the signals and analyze the geometry of the latent space

$$\boldsymbol{z}(t) = \mathcal{L}\{\boldsymbol{x}(t)\}$$ (e.g. Fourier power spectral density)

Obtain the **statistics** of the observers in short windows (assuming local **stationarity**)

$$\widehat{\boldsymbol{\mu}}(t) = \frac{1}{L_0} \sum_{\tau \in \mathcal{T}_t} \boldsymbol{z}(\tau) \qquad \left(L_0 = |\mathcal{T}_t|\right)$$

$$\widehat{\boldsymbol{C}}(t) = \frac{1}{L_0} \sum_{\tau \in \mathcal{T}_t} \Big(\boldsymbol{z}(\tau) - \widehat{\boldsymbol{\mu}}(t)\Big)\Big(\boldsymbol{z}(\tau) - \widehat{\boldsymbol{\mu}}(t)\Big)^T$$

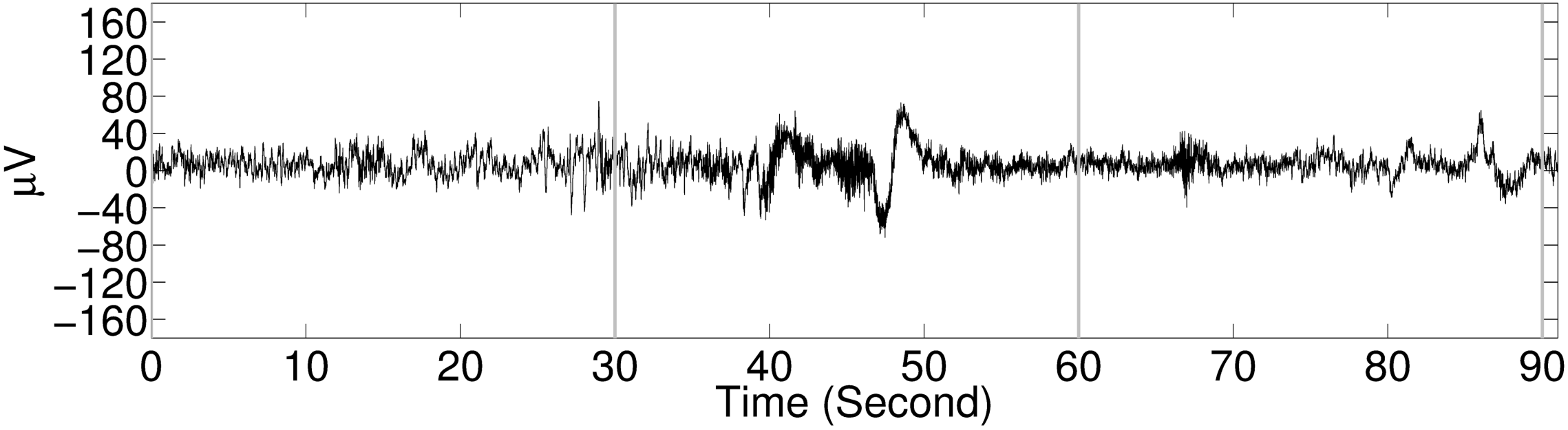
Define a **Mahalanobis** distance between samples

$$d^2(\boldsymbol{x}(t), \boldsymbol{x}(s)) = \frac{1}{2}\Big(\boldsymbol{z}(t) - \boldsymbol{z}(s)\Big)^T \Big(\boldsymbol{C}^\dagger(t) + \boldsymbol{C}^\dagger(s)\Big)\Big(\boldsymbol{z}(t) - \boldsymbol{z}(s)\Big)$$

Hau-tieng Wu, Ronen Talmon, and Yu-Lun Lo

Assess sleep stage by modern signal processing techniques

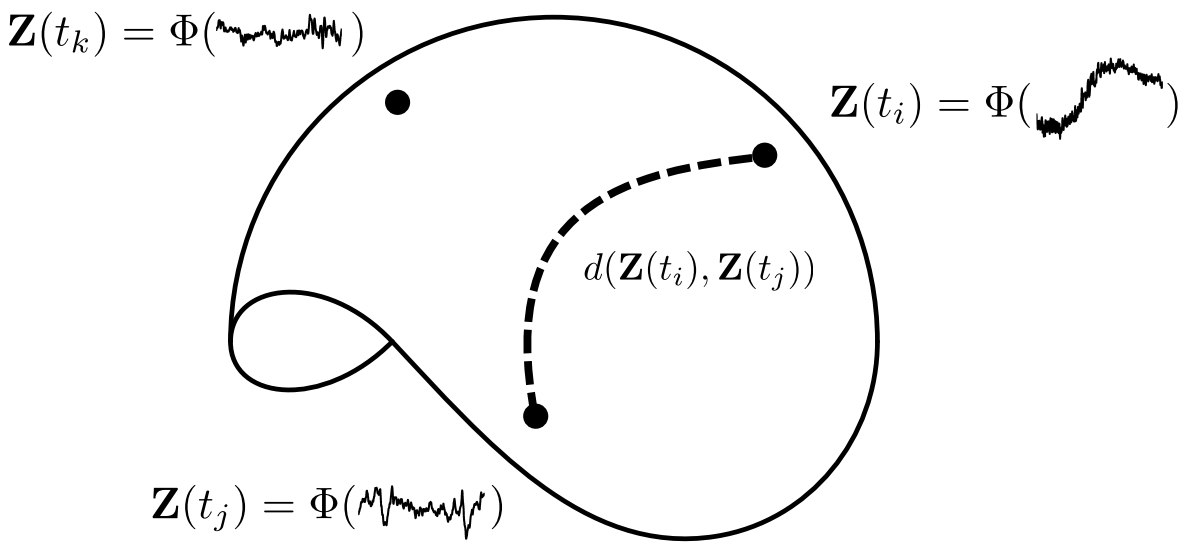
IEEE Transactions on Biomedical Engineering (2015)

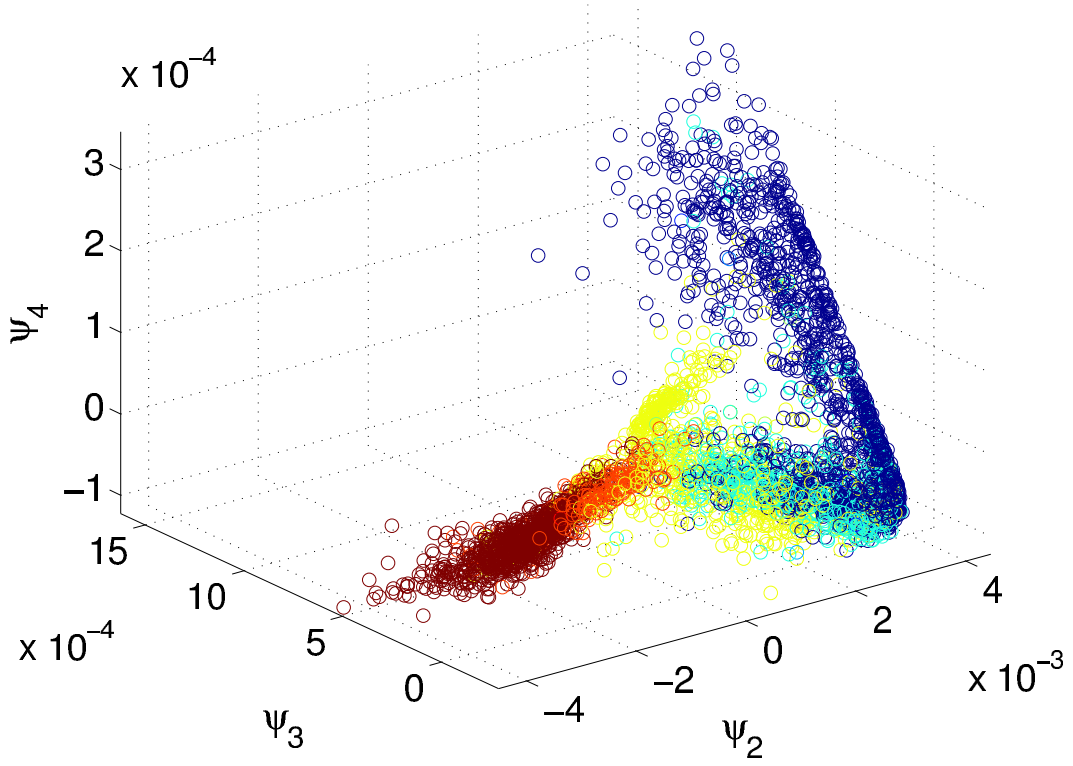


Hau-tieng Wu, Ronen Talmon, and Yu-Lun Lo

Assess sleep stage by modern signal processing techniques

IEEE Transactions on Biomedical Engineering (2015)

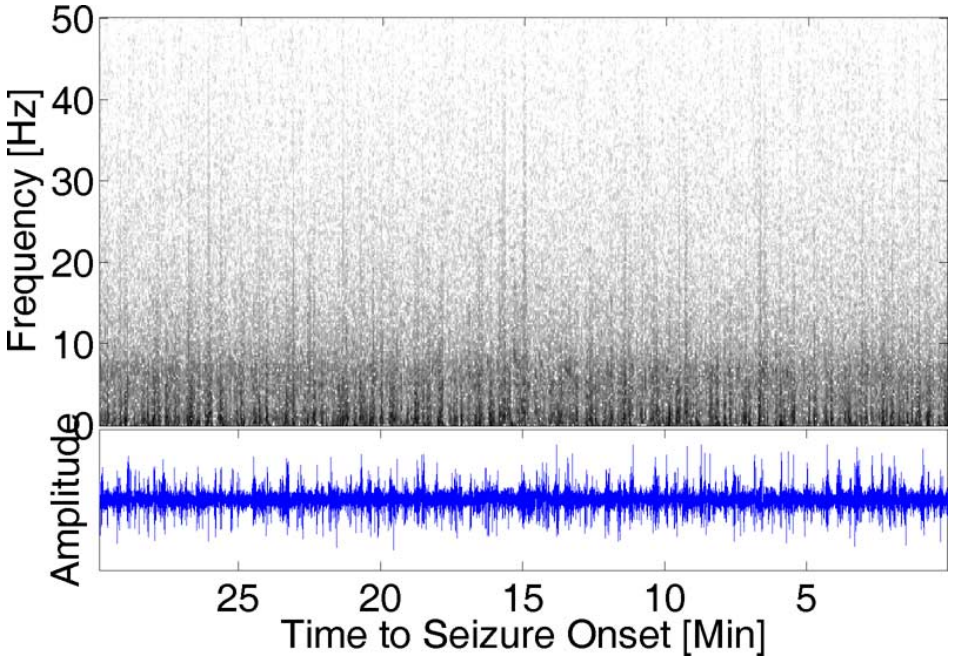


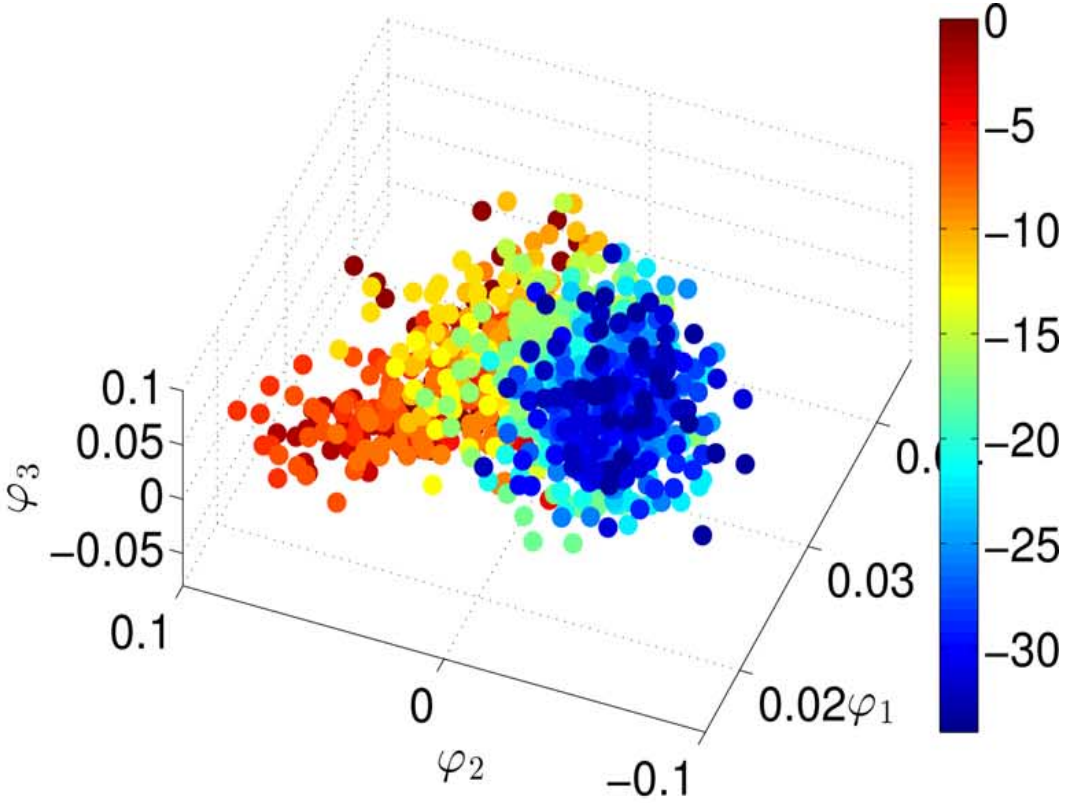


Ronen Talmon, Stéphane Mallat, Hitten Zavari, Ronald R. Coifman

Manifold Learning for Latent Variable Inference in Dynamical Systems

IEEE Transactions in Signal Processing (2015)

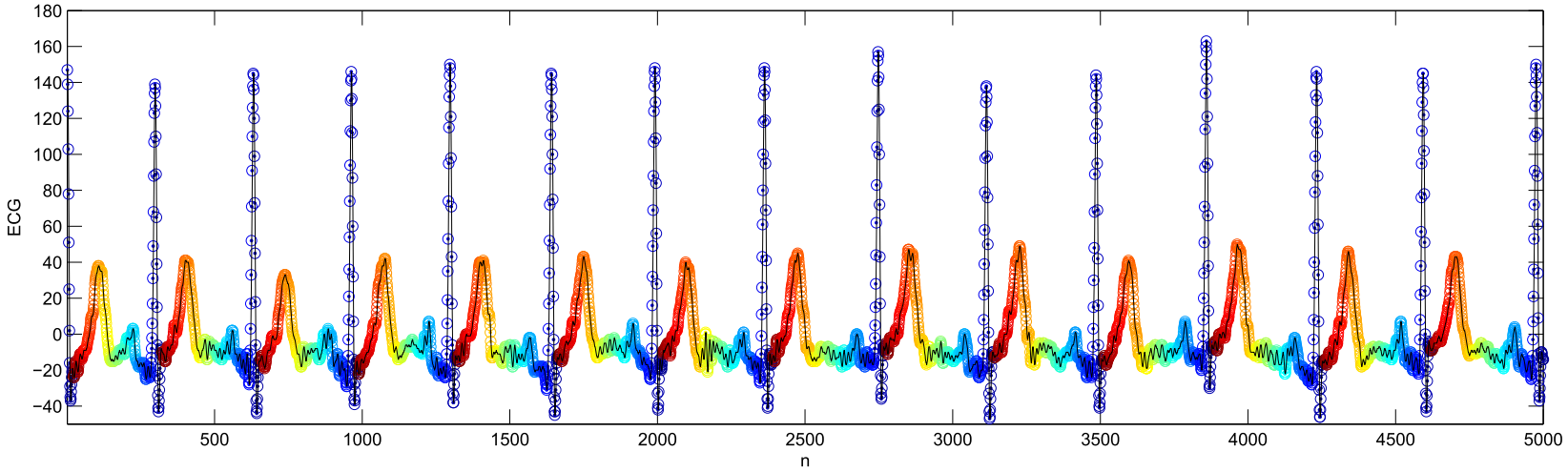


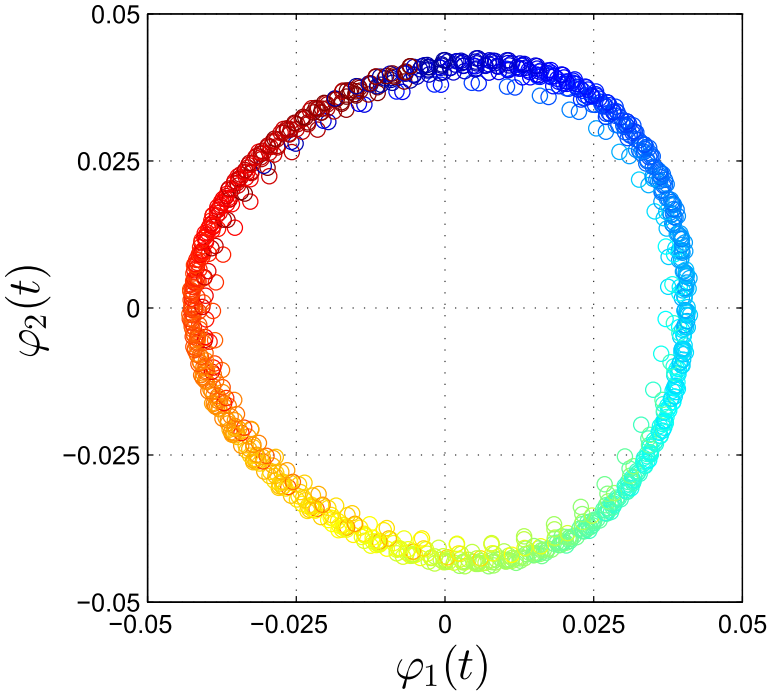


Jeremias Sulam, Yaniv Romano, and Ronen Talmon

Dynamical system classification with diffusion embedding for ECG-based person identification

Elsevier Signal Processing (2017)





Concluding remarks

- Diffusion maps and Laplacian eigenmaps are very **similar** techniques
- Tools from multiple **areas**: functional analysis, probability theory, physics, etc.
- Works dealing with large-scale implementation (**quadratic** complexity)
- Questions of **multi-modality** explored recently (Talmon and Wu, 2016)
- Interesting **extensions** to representation learning

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

Hope to see some of you again…

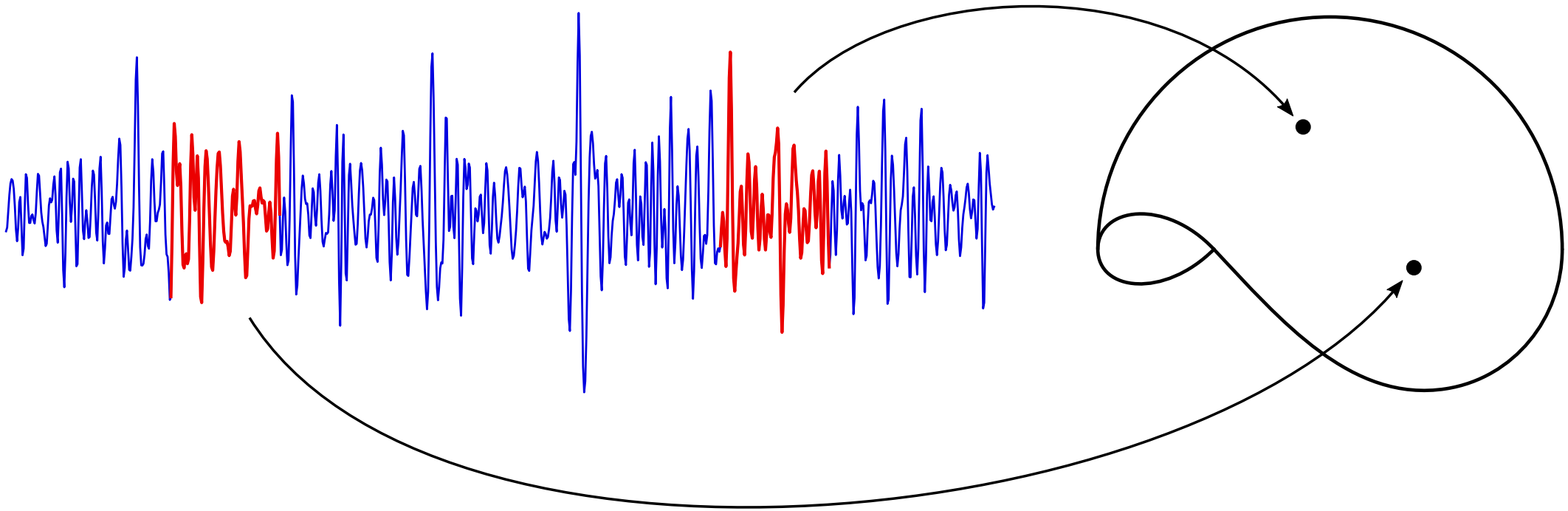
CHESS meeting
Monday, 3th April

Some results on
EEG signal analysis

An introduction to diffusion maps

and their use in time series analysis

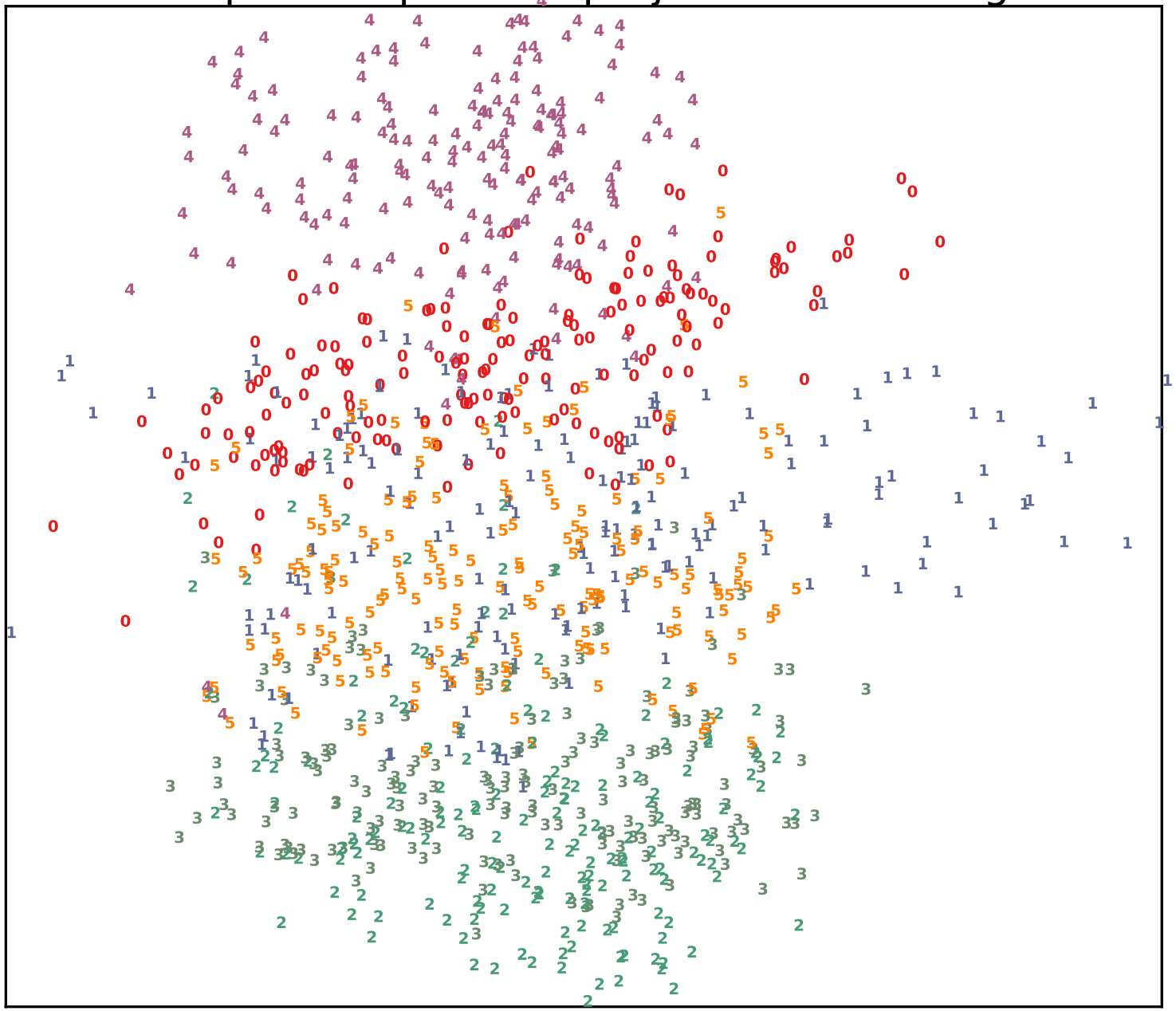
Pedro Rodrigues – VIBS team meeting, 17th March 2017



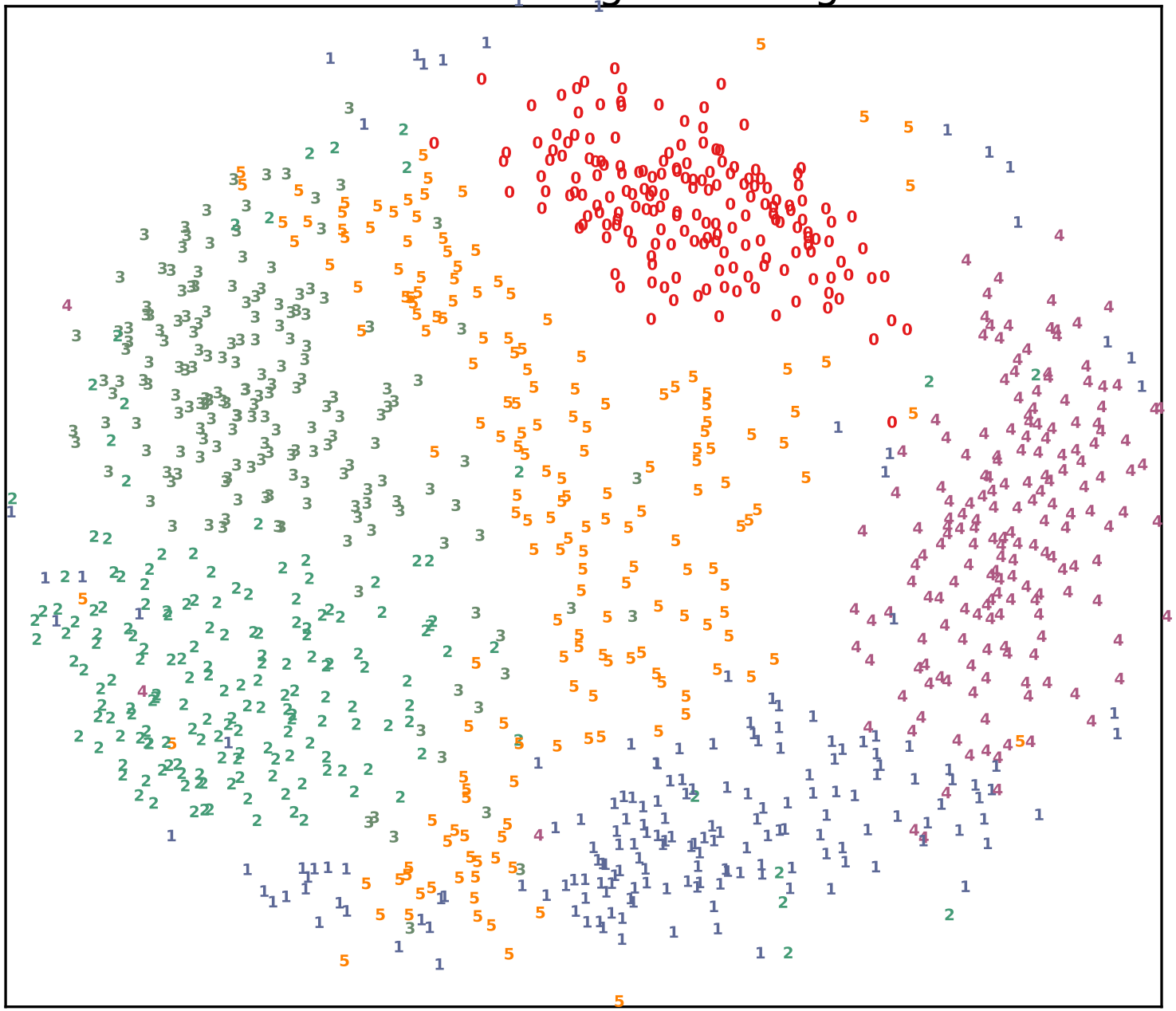
Example with a subset of the MNIST database

- Handwritten digits
- 400 samples
- 64 x 64 pixel images

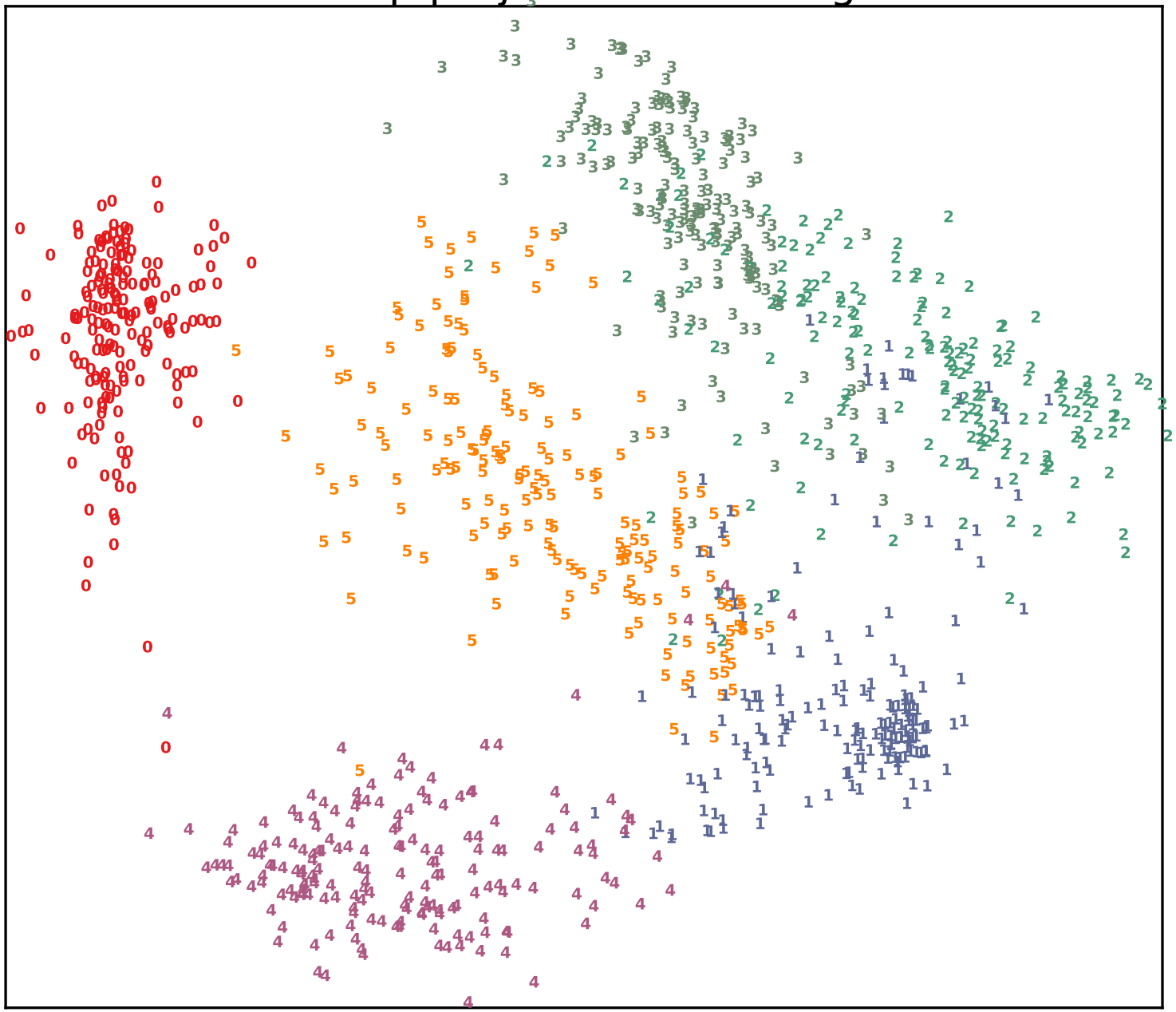
Principal Components projection of the digits



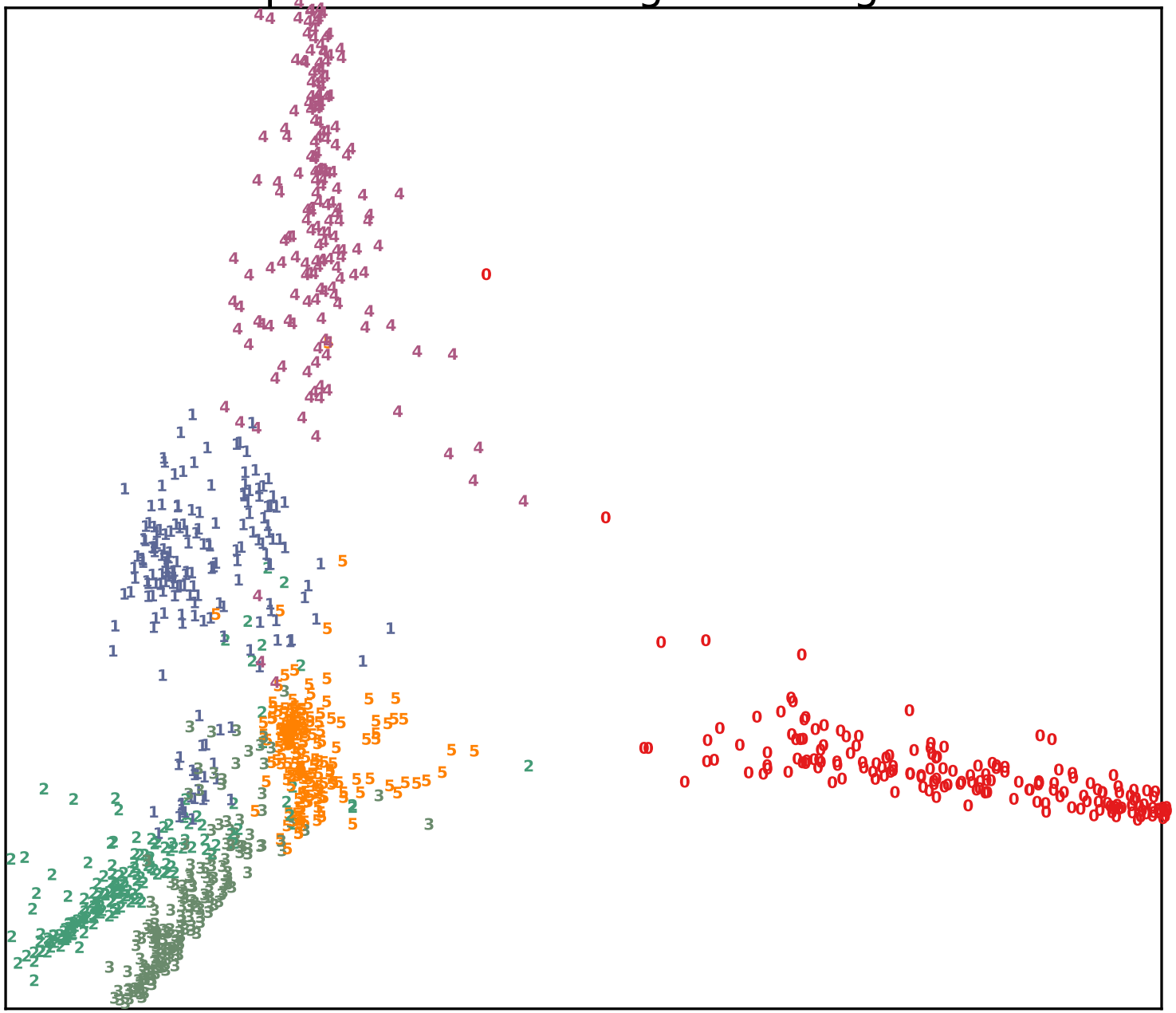
MDS embedding of the digits

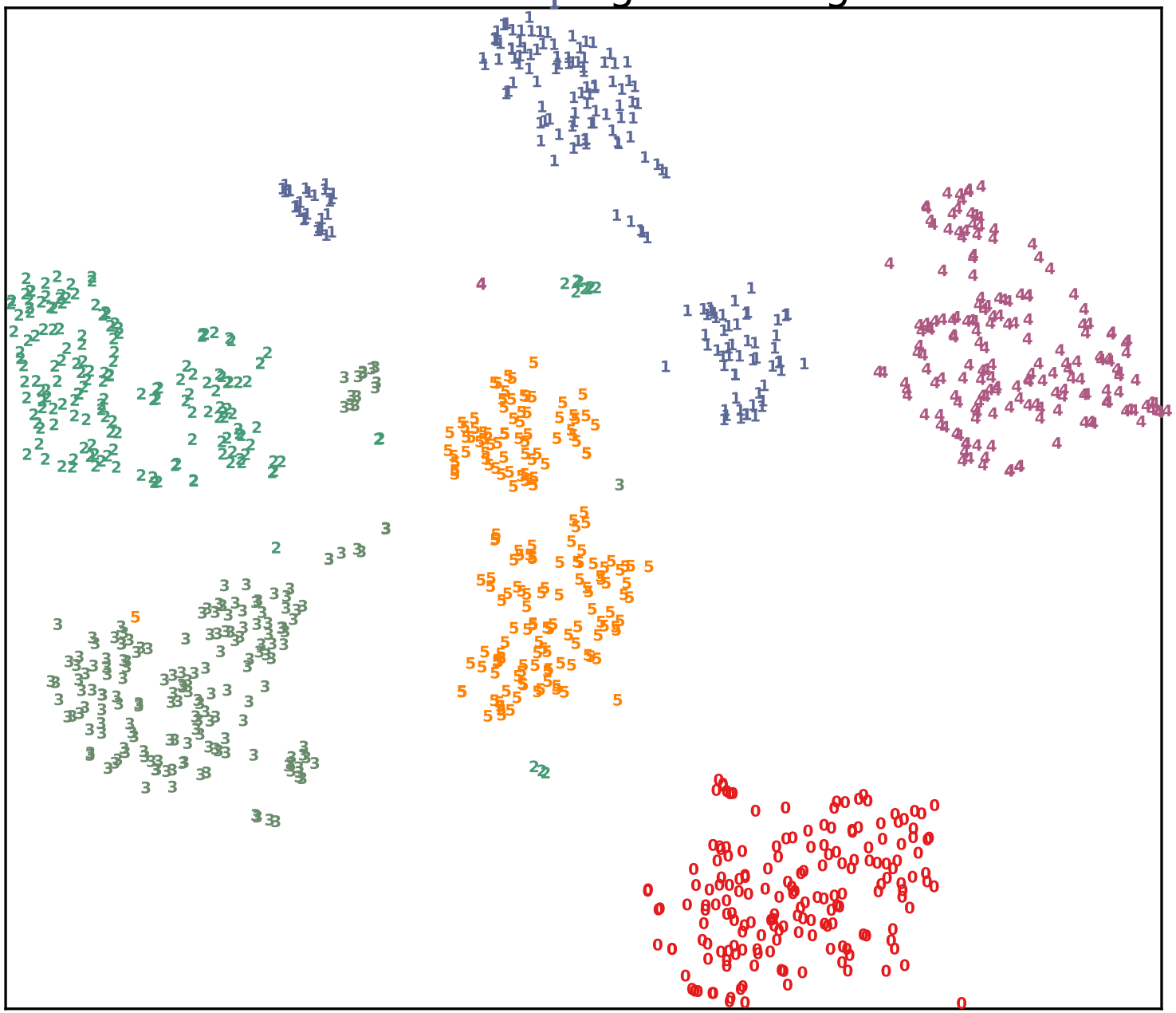


Isomap projection of the digits



Spectral embedding of the digits




t-SNE embedding of the digits