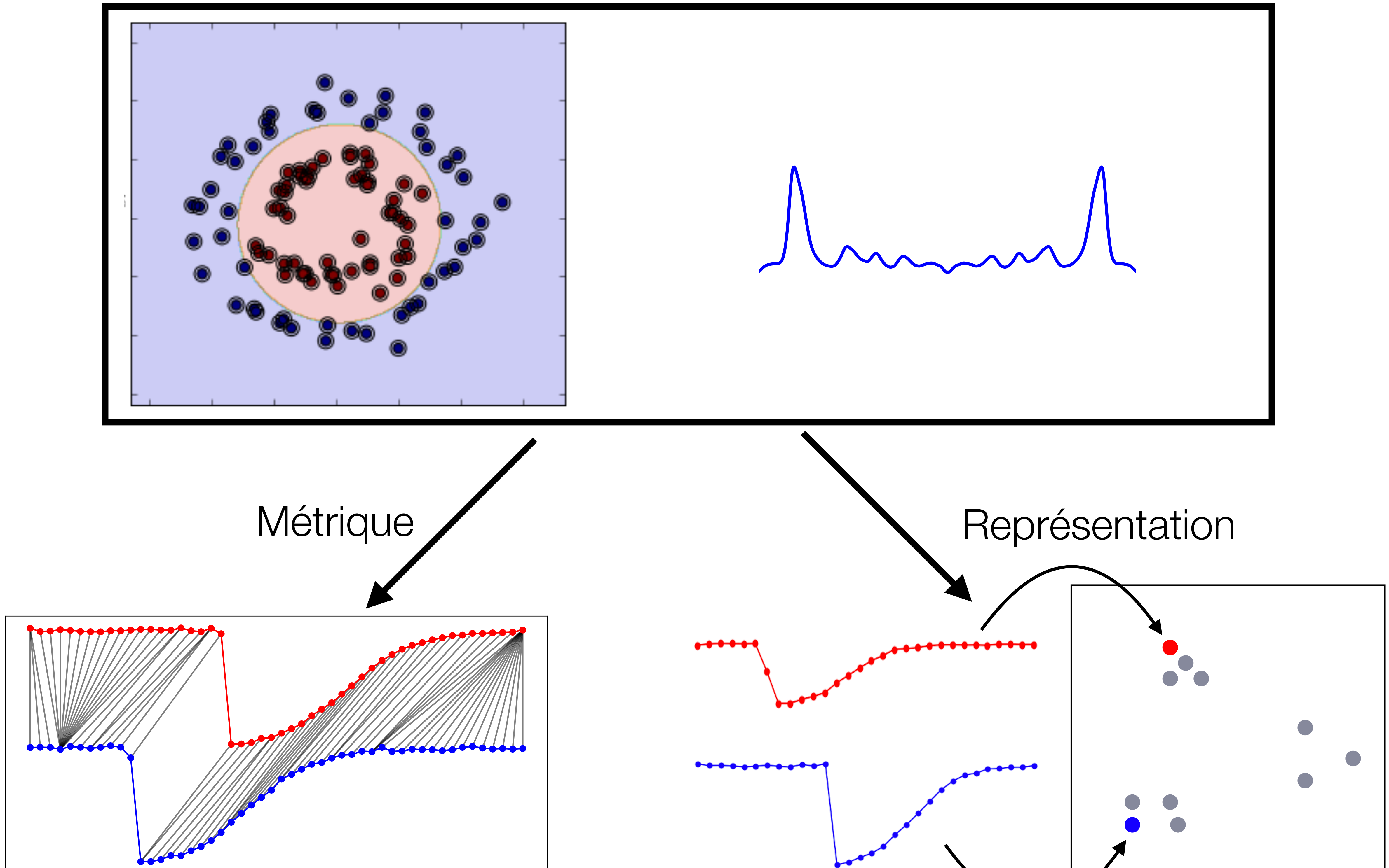
Machine Learning pour les séries temporelles en Python

Partie I : comparer des séries temporelles

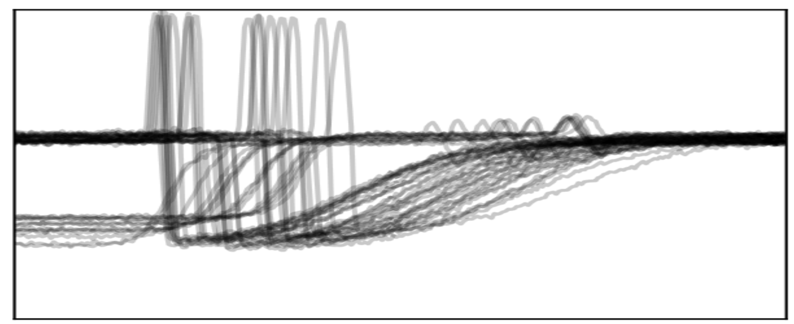
Yann Cabanes, Johann Faouzi, Romain Tavenard

Tutoriel présenté à CAp 2023
tslearn-team.github.io/tutoriel-cap2023/

ML pour les séries temporelles



Comparer des séries temporelles
Pourquoi utiliser des métriques dédiées ?



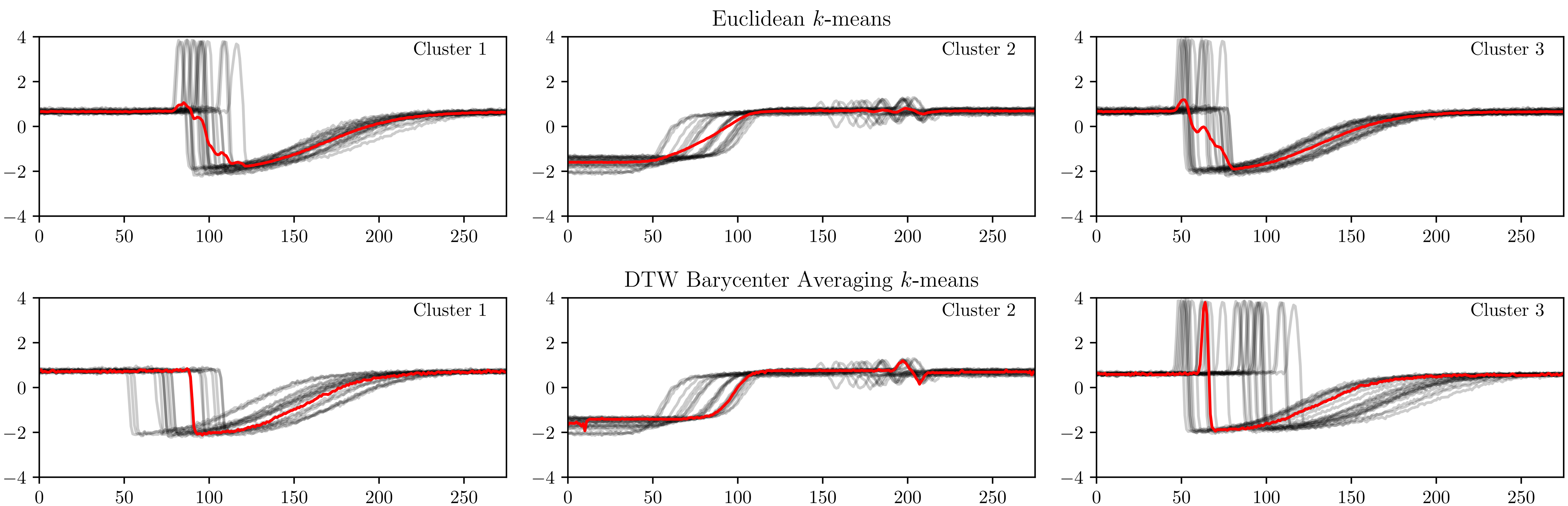
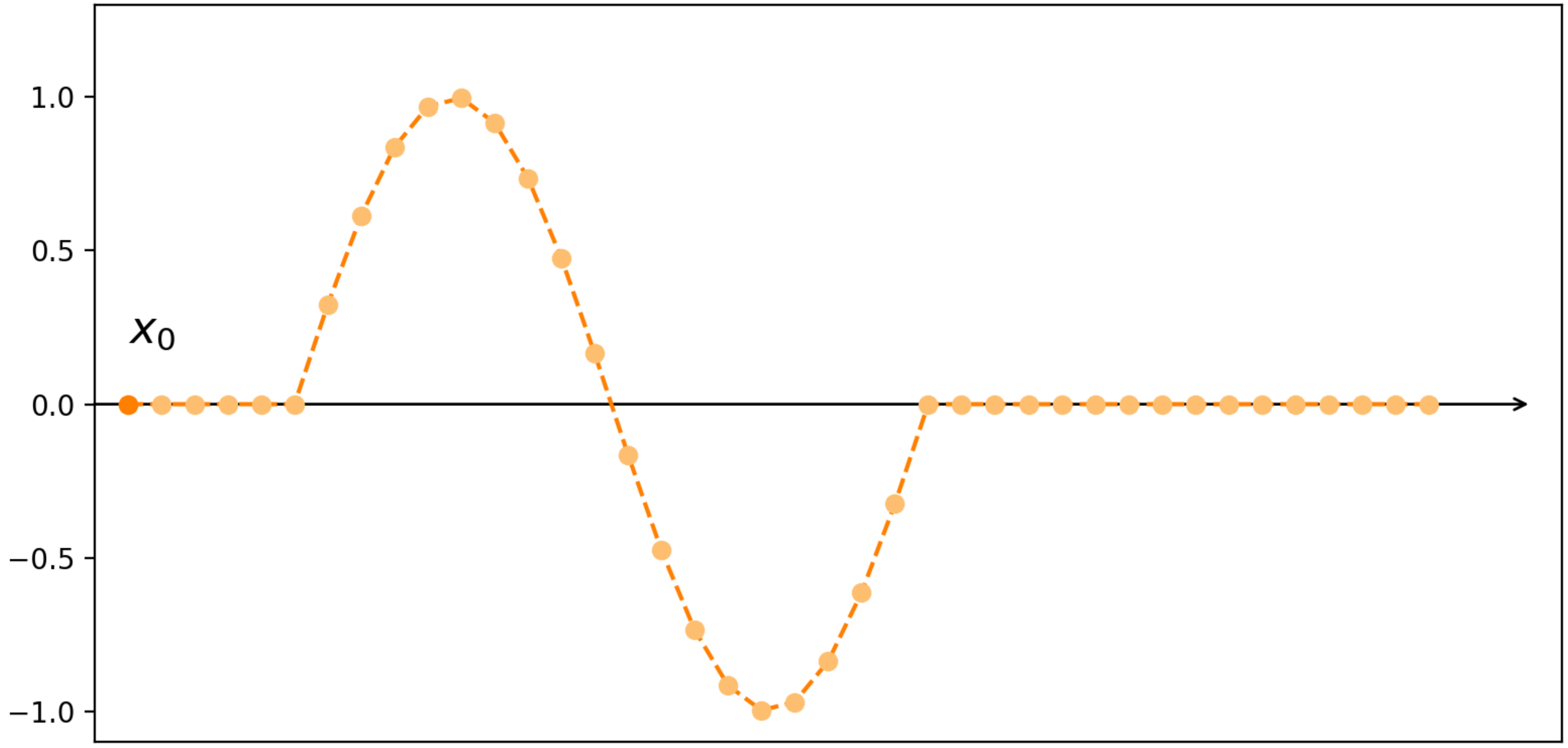


Illustration issue des docs tslearn

Comparer des séries temporelles
Ce qu’on entend par « série temporelle »

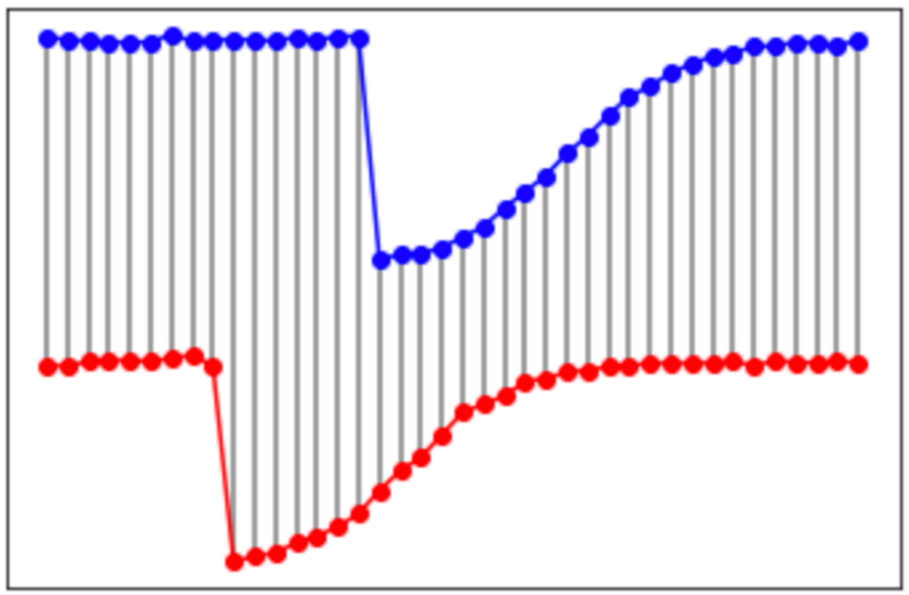
$$x = (x_0, \ldots, x_{n-1})$$



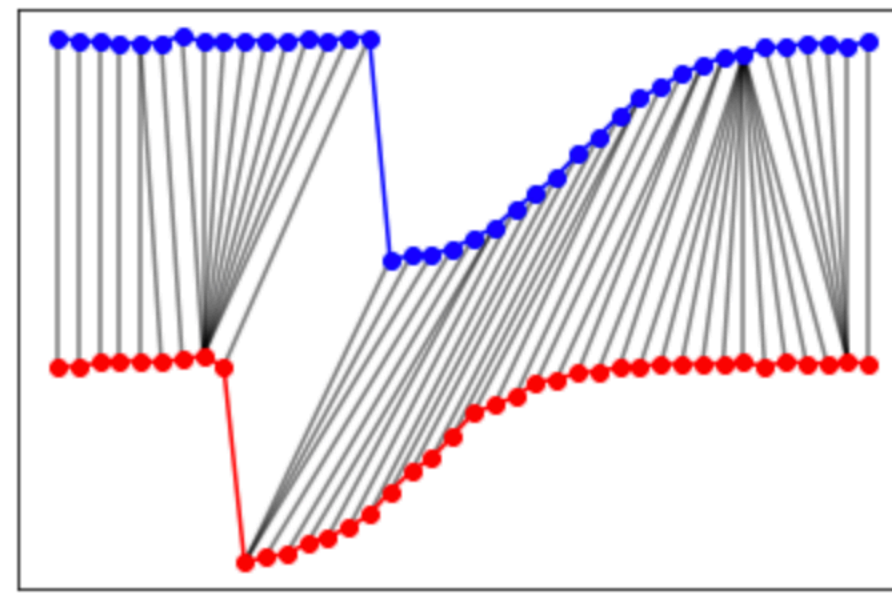
Y. Cabanes, J. Faouzi, R. Tavenard - tslearn-team.github.io/tutoriel-cap2023/

Comparer des séries temporelles
Dynamic Time Warping (DTW) : intuition

H. Sakoe

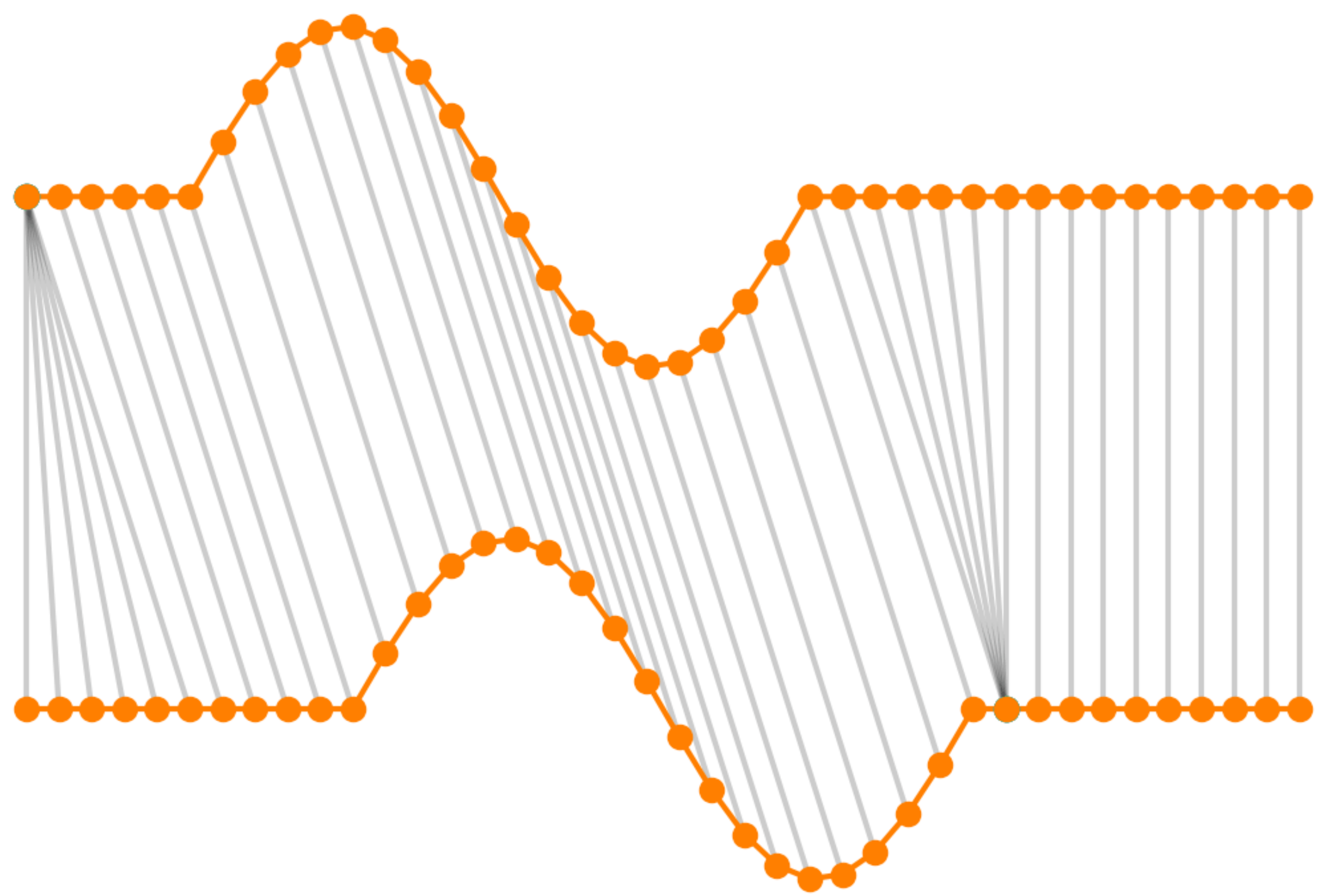


Appariement basé distance
Euclidienne

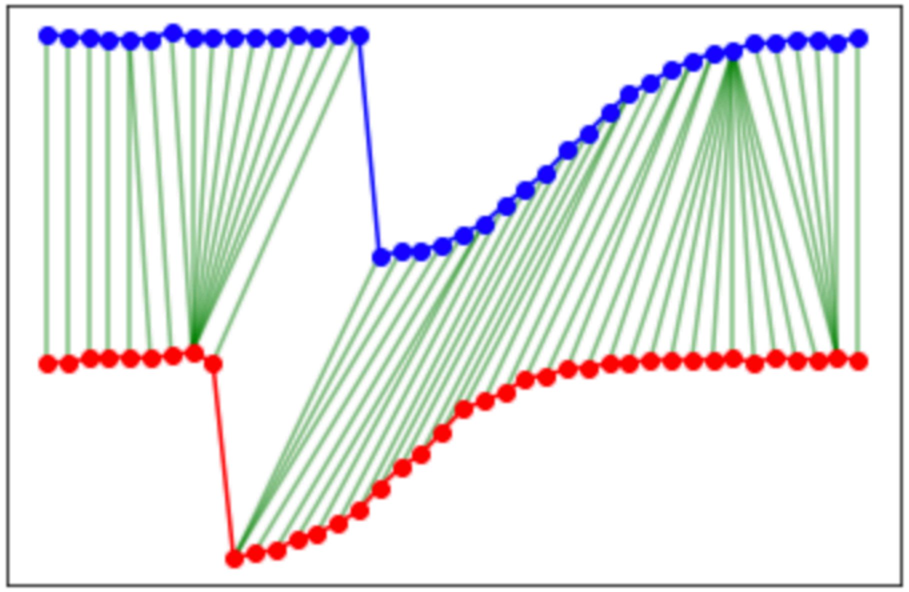


Appariement basé
Dynamic Time Warping

Comparer des séries temporelles
Dynamic Time Warping (DTW) : intuition



Comparer des séries temporelles
Dynamic Time Warping (DTW)



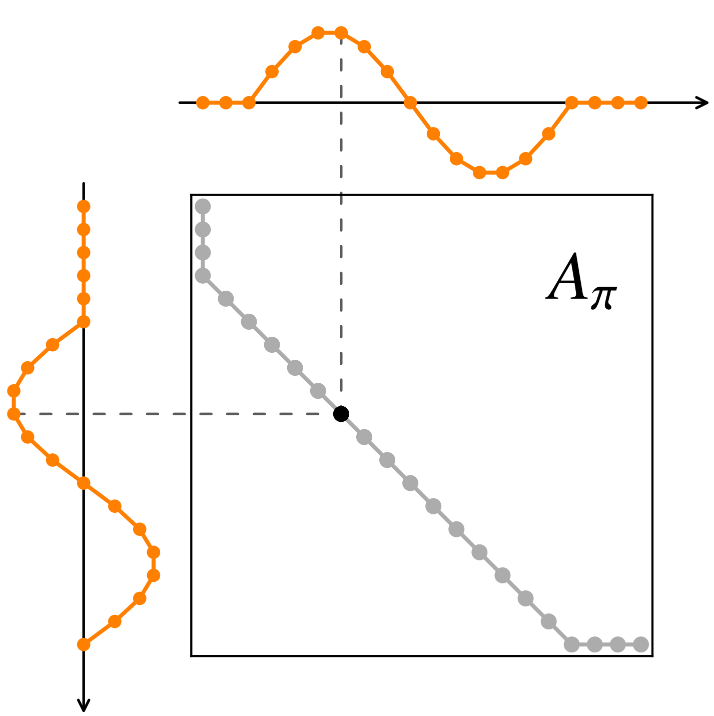
- DTW Optimization problem

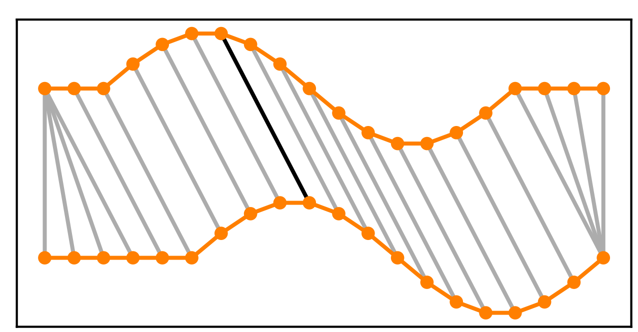
$$DTW_q(\mathbf{x}, \mathbf{x}') = \min_{\pi \in \mathcal{A}(\mathbf{x}, \mathbf{x}')} \left(\sum_{(i,j) \in \pi} d(x_i, x'_j)^q \right)^{\frac{1}{q}}$$

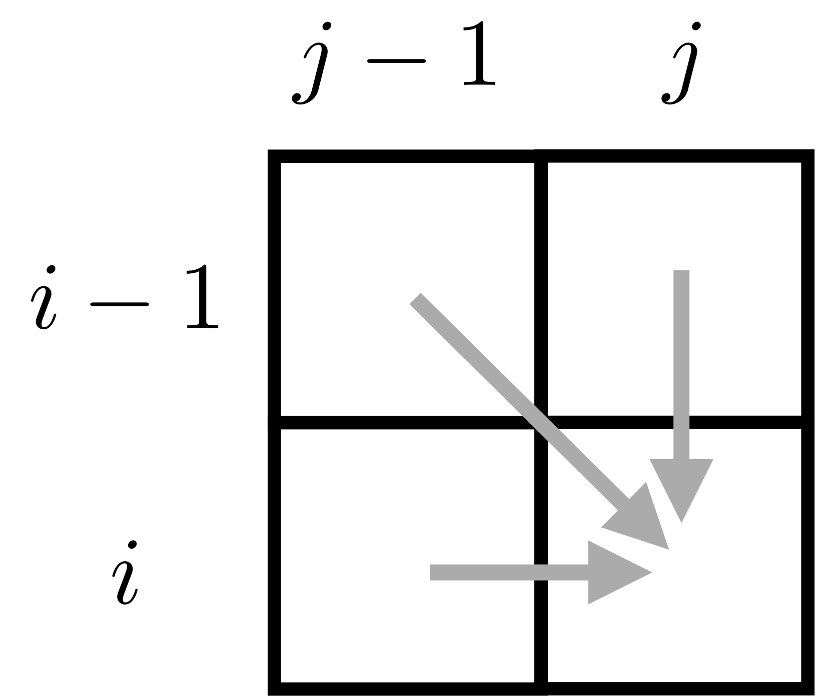
- Optimization on the path π
  1. Should match beginning (resp. end) of time series
  2. Should be monotonically increasing
  3. Should not skip elements

Comparer des séries temporelles
Dynamic Time Warping (DTW)

- Optimization on the path π
    1. Should match beginning (resp. end) of time series
    2. Should be monotonically increasing
    3. Should not skip elements







Comparer des séries temporelles
Soft-DTW

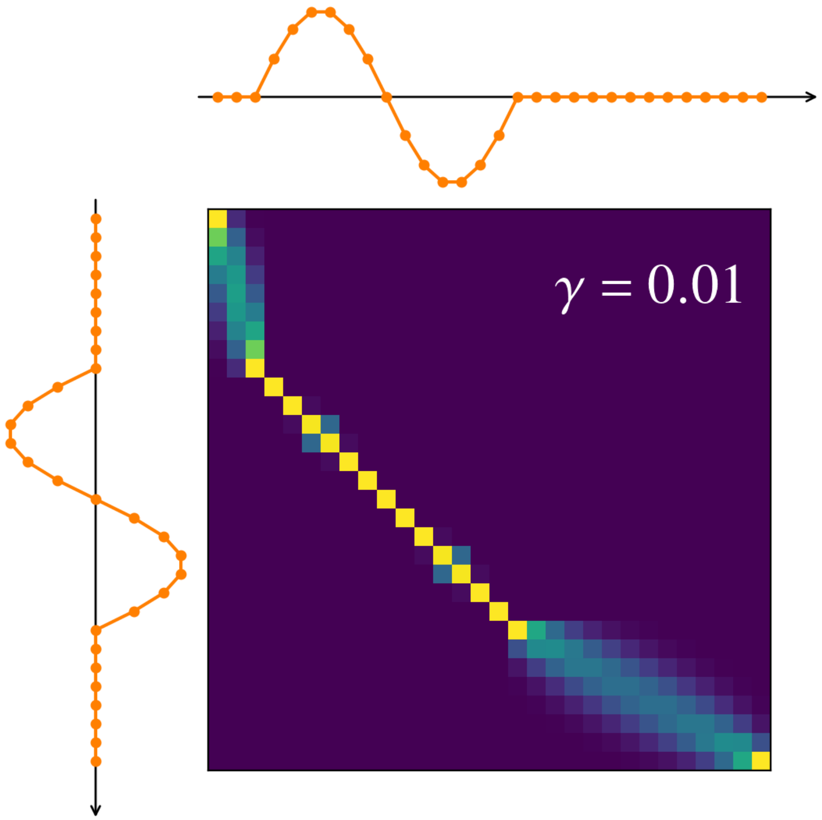
M. Cuturi

M. Blondel

- DTW Optimization problem

$$DTW_q(\mathbf{x}, \mathbf{x}') = \min_{\pi \in \mathcal{A}(\mathbf{x}, \mathbf{x}')} \left(\sum_{(i,j) \in \pi} d(x_i, x'_j)^q \right)^{\frac{1}{q}}$$

- Differentiable variant: soft-DTW
  - Use $\text{soft-}\min^{\gamma}$ in place of $\min$



Comparer des séries temporelles
Un k-means à la sauce DTW

Algorithm 3: Lloyd's algorithm (k-means)

Data: $(\mathbf{x}^1, \dots \mathbf{x}^n)$: time series dataset

$\{\mathbf{b}^1, \dots, \mathbf{b}^k\} \leftarrow \text{PickFrom}(\mathbf{x}^1, \dots \mathbf{x}^n)$ `// Or use k-means++ init`

for $e = 1..n_{iter}$ **do**

  for $i = 1..n$ **do**

    $a_i \leftarrow \text{NearestNeighborIndex}(\mathbf{x}^i, \{\mathbf{b}^1, \dots, \mathbf{b}^k\})$

  end

  for $j = 1..k$ **do**

    $\mathbf{b}^j \leftarrow \text{Barycenter}(\{\mathbf{x}^i | a_i = j\})$

  end

end

return z

Comparer des séries temporelles
softDTW comme fonction de coût

- Tâche
  - Prédiction du futur d'une série temporelle
- Outil
  - Réseau de neurone
- Deux possibilités

$$\mathcal{L}_{\mathrm{MSE}}(\mathbf{y}, \hat{\mathbf{y}}) = \sum_{i=1}^{T} d(y_i, \hat{y}_i)^2$$

$$\mathcal{L}_{\text{soft-}DTW^\gamma}(\mathbf{y}, \hat{\mathbf{y}}) = \text{ soft-} \min_{\pi \in \mathcal{A}(T,T)}{}^{\gamma} \sum_{(i,j) \in \pi} d(y_i, \hat{y}_j)^2$$

Y. Cabanes, J. Faouzi, R. Tavenard - tslearn-team.github.io/tutoriel-cap2023/

Comparer des séries temporelles
softDTW comme fonction de coût

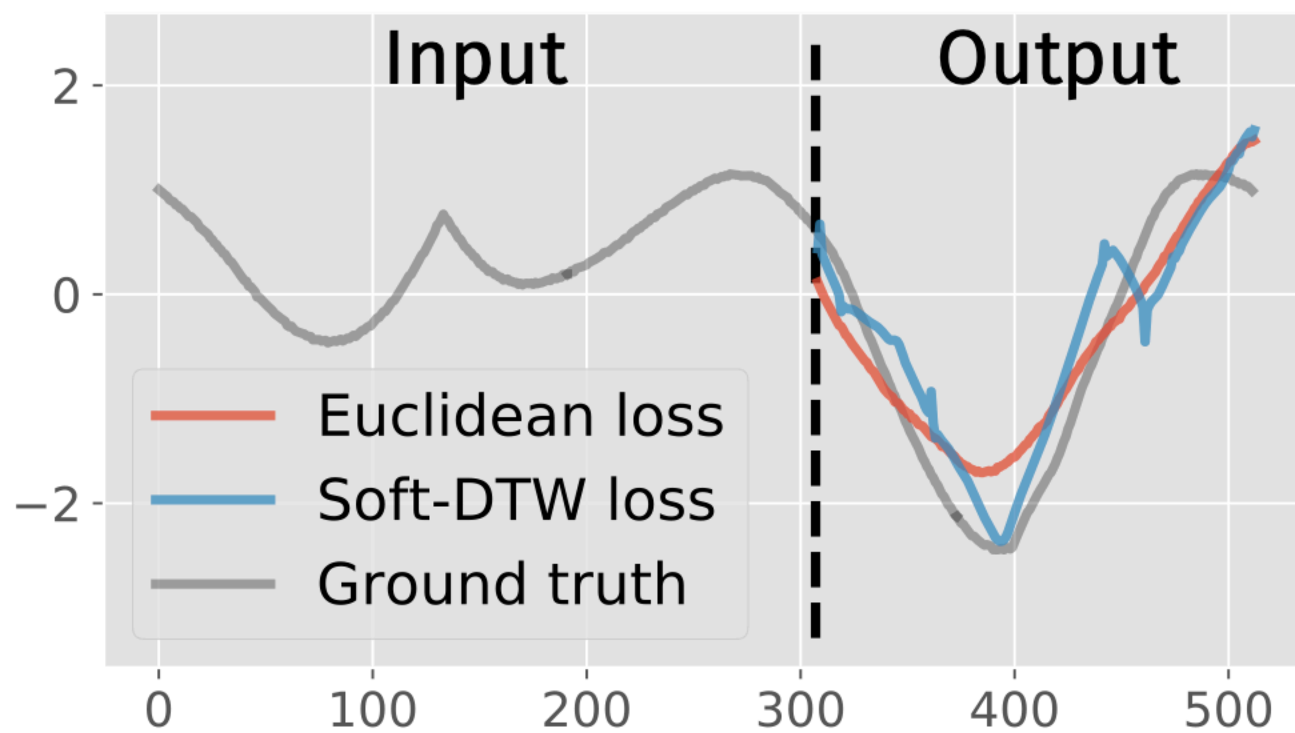


Illustration issue de [Cuturi & Blondel, 2017]

tslearn: Présentation en 1 diapo

- Installation

pip install tslearn
conda install -c conda-forge tslearn

- Utilisation : à la scikit-learn

```
>>> from tslearn.datasets import UCR_UEA_datasets
>>> from tslearn.clustering import TimeSeriesKMeans
>>>
>>> X_train, y_train, X_test, y_test = UCR_UEA_datasets().load_dataset("TwoPatterns")
>>> print(X_train.shape)
(1000, 128, 1)
>>>
>>> km = TimeSeriesKMeans(n_clusters=3, metric="dtw")
>>> km.fit(X_train)
```