ETH zürich

A Wasserstein Subsequence Kernel for Time Series

Christian Bock, Matteo Togninalli, Elisabetta Ghisu, Thomas Gumbsch, Bastian Rieck, Karsten Borgwardt

Department of Biosystems Science and Engineering
Machine Learning and Computational Biology Group

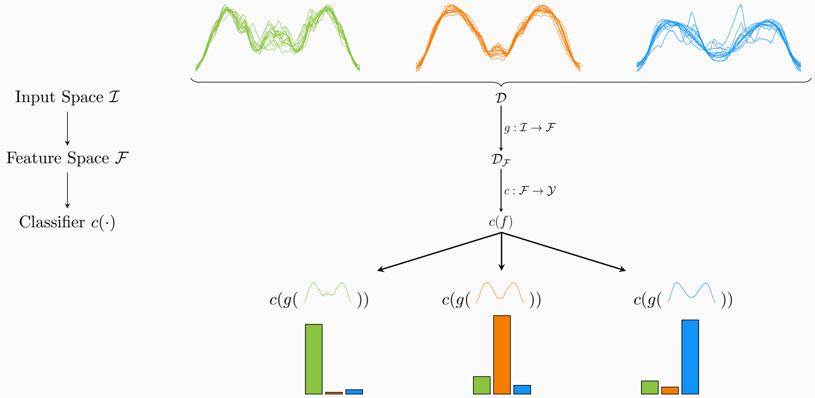
ETH Colors

ETHa ETHb ETHc ETHd ETHe ETHf ETHg ETHh ETHi

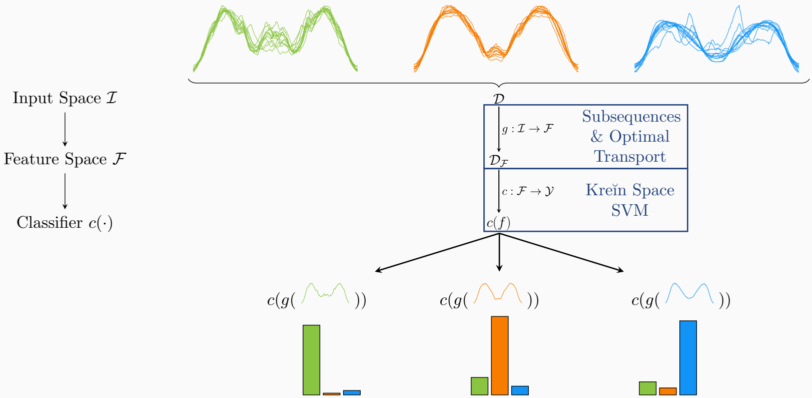
Table of contents (internal)

Motivation

Time Series Classification in a Nutshell



Time Series Classification in a Nutshell

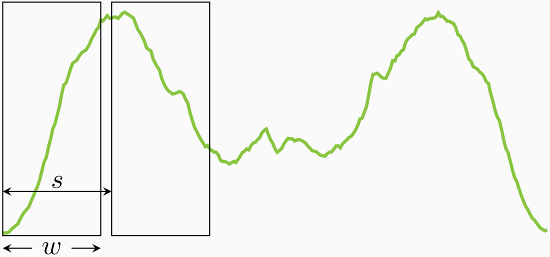


Feature Space: Subsequences and Optimal Transport

Subsequence Extraction

The Sliding Window Approach

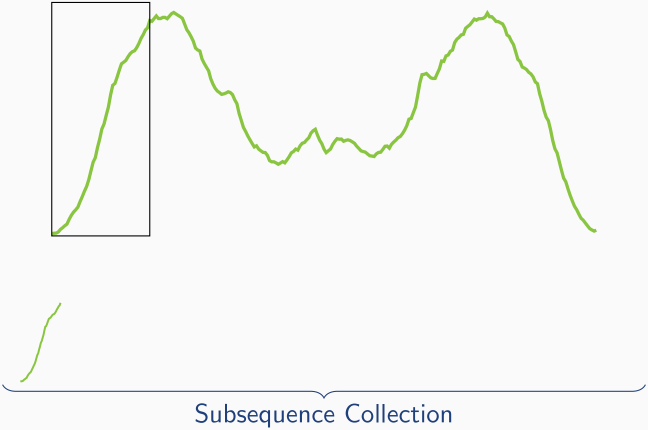
- Window Size w
- Stride s



Subsequence Extraction

The Sliding Window Approach

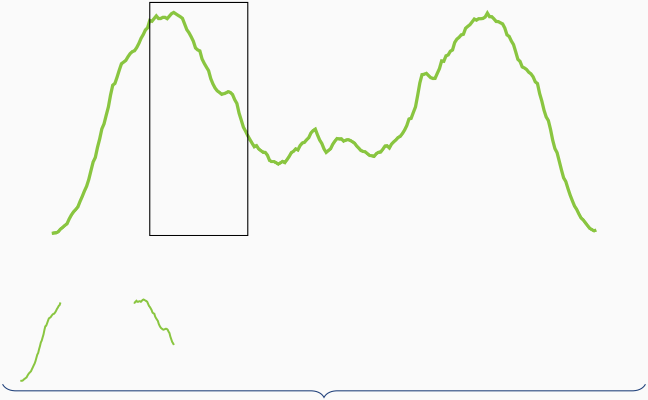
- Window Size w
- Stride s



Subsequence Extraction

The Sliding Window Approach

- Window Size w
- Stride s

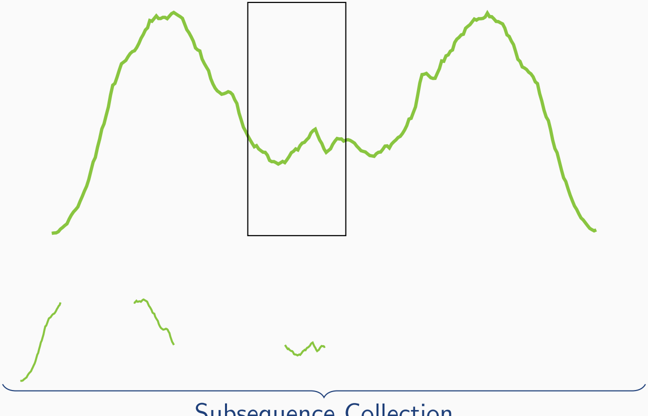


Subsequence Collection

Subsequence Extraction

The Sliding Window Approach

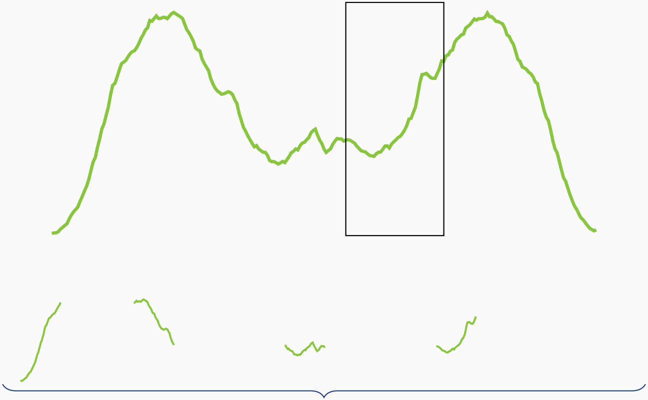
- Window Size w
- Stride s



Subsequence Extraction

The Sliding Window Approach

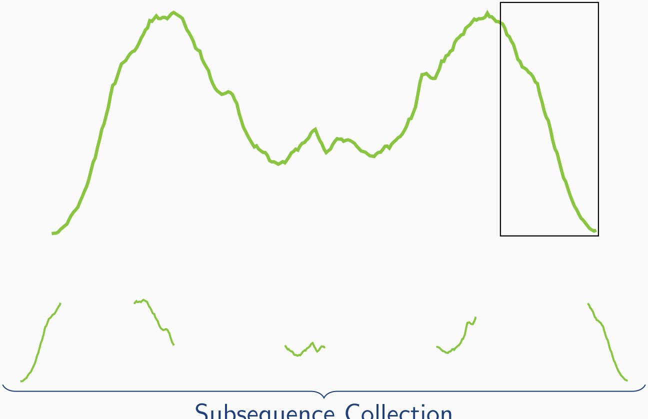
- Window Size w
- Stride s



Subsequence Extraction

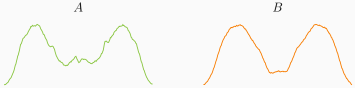
The Sliding Window Approach

- Window Size w
- Stride s

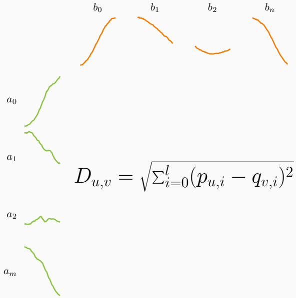


Subsequence Distance Matrix

Given Two Time Series



we Construct the Subsequence Distance Matrix D

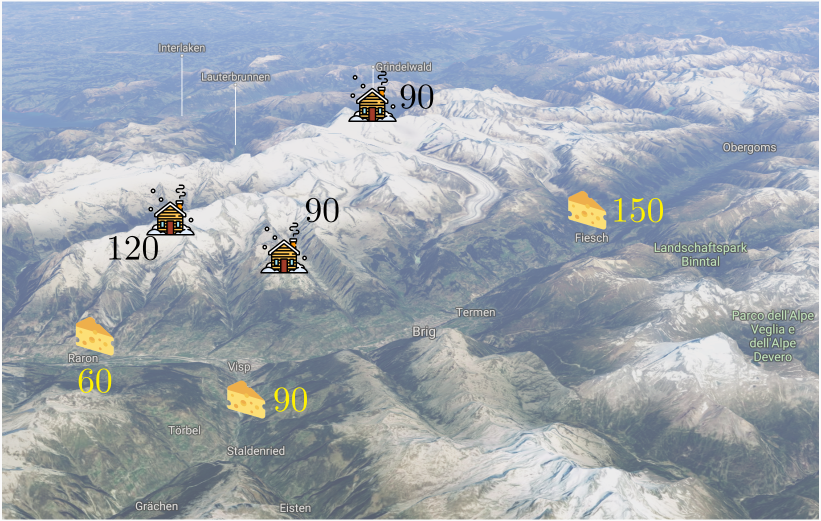


$$D_{u,v} = \sqrt{\Sigma_{i=0}^{l} (p_{u,i} - q_{v,i})^2}$$

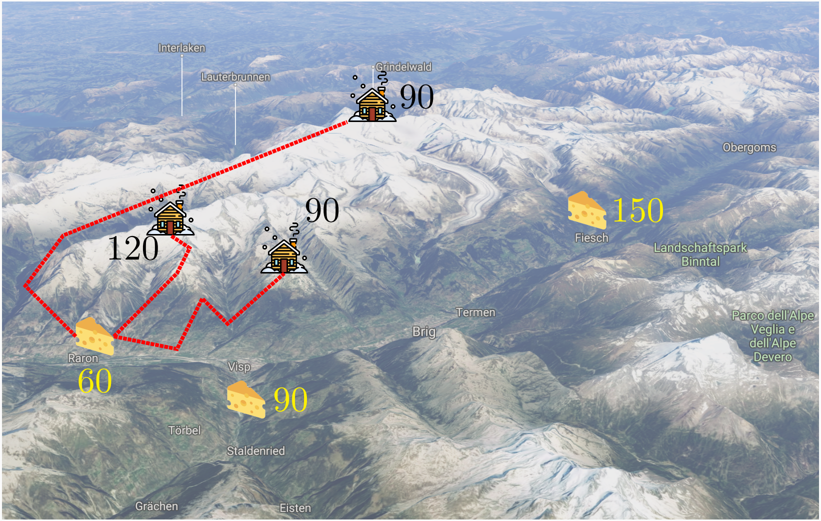
and obtain $\quad W_1 (A, B) = \min_{P \in \Gamma(A,B)} \langle D, P \rangle_{\mathrm{F}}$

Optimal Transport and the Wasserstein Distance

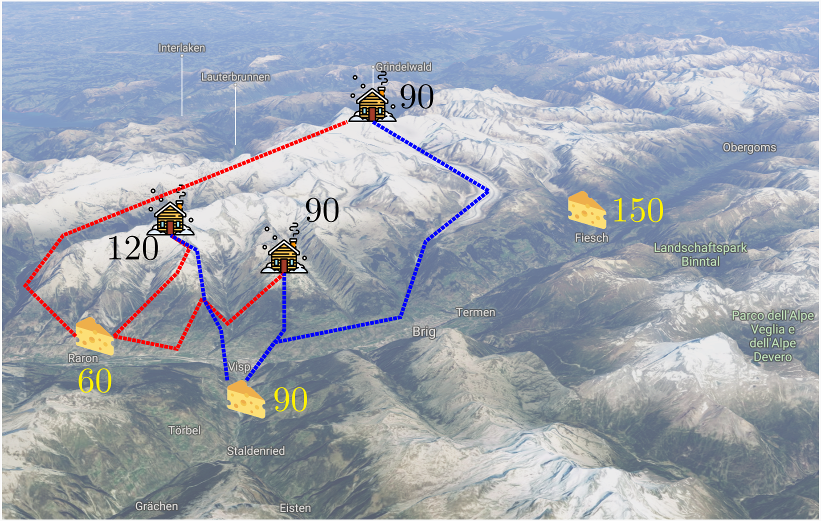
The Swiss Kantorovich Problem



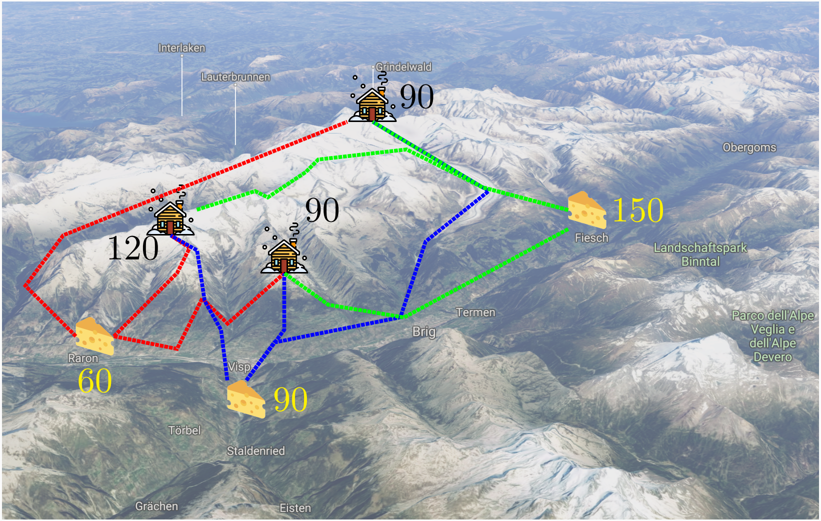
The Swiss Kantorovich Problem



The Swiss Kantorovich Problem

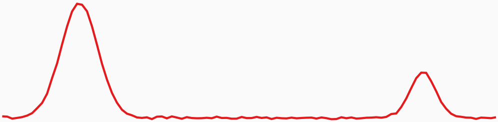


The Swiss Kantorovich Problem

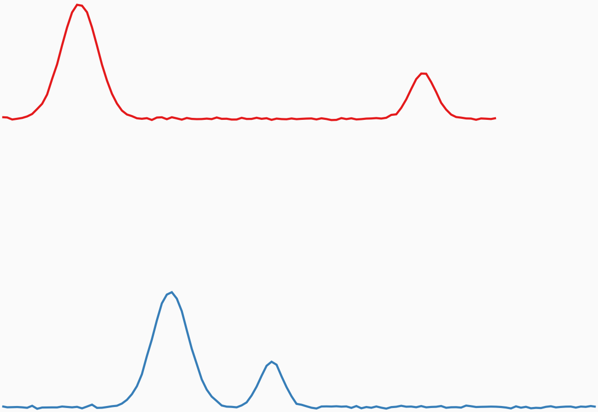


A Wasserstein Kernel for Time Series

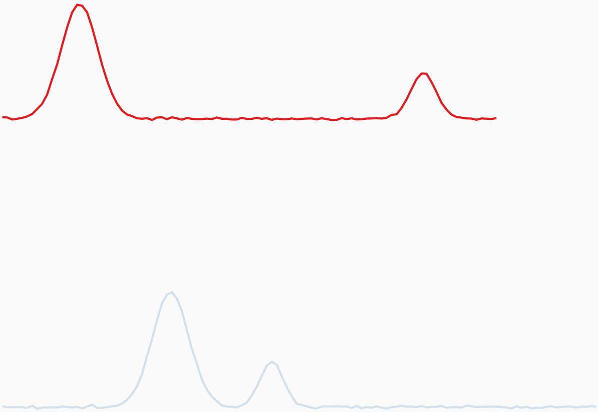
Time Series Pairwise Distance Calculation



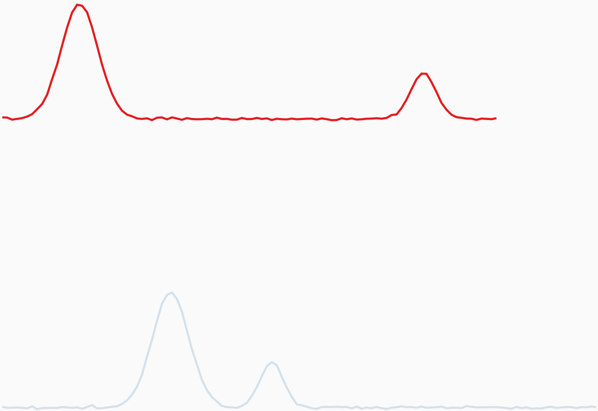
Time Series Pairwise Distance Calculation



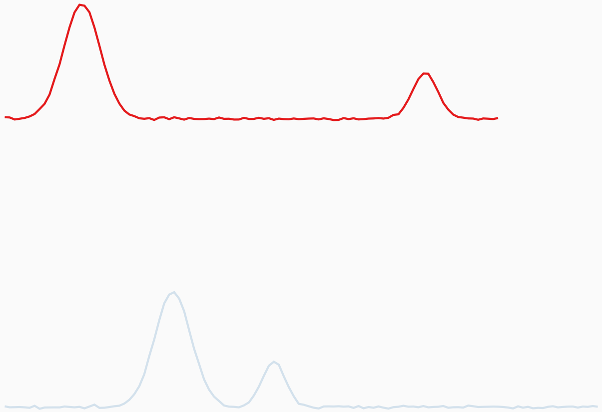
Time Series Pairwise Distance Calculation



Time Series Pairwise Distance Calculation



Time Series Pairwise Distance Calculation



Wasserstein Distance Matrix

From Distance to Kernels

Definition (Wasserstein time series kernel)

Let T_i and T_j be two time series, and $s_{i1}, \ldots, s_{iU}$ as well as $s_{j1}, \ldots, s_{jV}$ be their respective subsequences. Moreover, let D be a $U \times V$ matrix that contains the pairwise distances of all subsequences, such that $D_{uv} := \operatorname{dist}(s_{iu}, s_{jv})$, where $\operatorname{dist}(\cdot, \cdot)$ denotes the usual Euclidean distance. The optimisation problem

$$W_1(T_i, T_j) := \min_{P \in \Gamma(T_i, T_j)} \langle D, P \rangle_{\mathrm{F}}, \tag{1}$$

yields the optimal transport cost to transform T_i into T_j by means of their subsequences. Then, given $\lambda \in_{>0}$, we can define

$$\mathrm{WTK}(T_i, T_j) := \exp\left(-\lambda W_1(T_i, T_j)\right), \tag{2}$$

which we refer to as our *Wasserstein-based subsequence kernel*;

Wasserstein Time Series Kernel

Algorithm 1 Wasserstein Time Series Kernel

Input: Time series for training and testing $\mathcal{T}_{\text{train}}$, $\mathcal{T}_{\text{test}}$; subsequence length w; kernel weight factor λ

Output: $\mathcal{K}^{\text{train}}, \mathcal{K}^{\text{test}}$

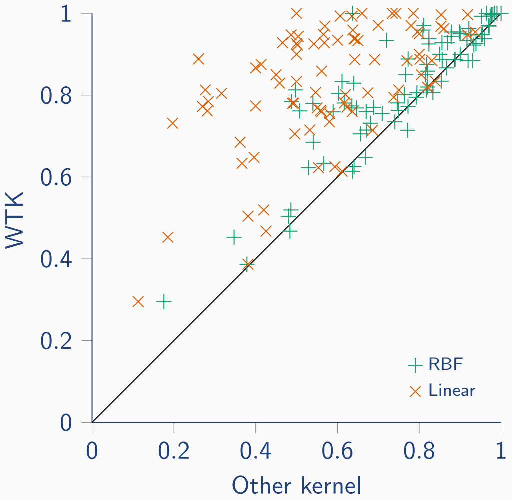
$\mathcal{S}^{\text{train}} \leftarrow \textsc{Subsequences}(\mathcal{T}_{\text{train}}, w)$ // Extract subsequences
$\mathcal{S}^{\text{test}} \leftarrow \textsc{Subsequences}(\mathcal{T}_{\text{test}}, w)$ // Extract subsequences
for $T_i \in \mathcal{T}_{\text{train}}$ **do**
    for $T_j \in \mathcal{T}_{\text{train}}$ **do**
        $\mathcal{D}_{ij}^{\text{train}} \leftarrow \mathrm{W}_1\left(\mathcal{S}_i^{\text{train}}, \mathcal{S}_j^{\text{train}}\right)$ // Wasserstein distance calculation (train)
    end for
    for $T_k \in \mathcal{T}_{\text{test}}$ **do**
        $\mathcal{D}_{ik}^{\text{test}} \leftarrow \mathrm{W}_1\left(\mathcal{S}_i^{\text{train}}, \mathcal{S}_k^{\text{test}}\right)$ // Wasserstein distance calculation (test)
    end for
end for
$\mathcal{K}^{\text{train}} \leftarrow \exp\left(-\lambda \mathcal{D}^{\text{train}}\right)$ // Kernel matrix calculation
$\mathcal{K}^{\text{test}} \leftarrow \exp\left(-\lambda \mathcal{D}^{\text{test}}\right)$ // Kernel matrix calculation
return $\mathcal{K}^{\text{train}}, \mathcal{K}^{\text{test}}$

Results

Experiments

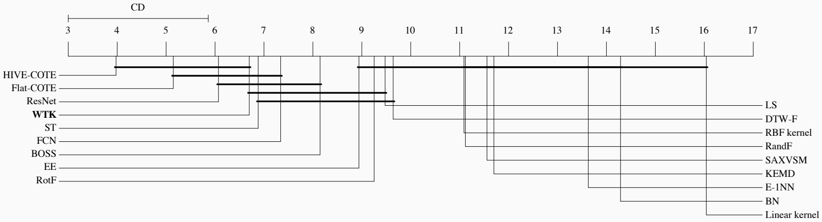
Datasets	*UCR Time Series Archive* 85 datasets predetermined train/test splits
Hyperparameters	Selected via a 5-fold cross-validation on the training set
Evaluation metric	Classification accuracy
Comparison partners	– Other kernels – DTW-1NN – State-of-the-art methods

Comparison with Other Kernels



Critical Difference Plot

The classification performances of methods sharing horizontal bars are not significantly different.



Take aways

Questions?

References