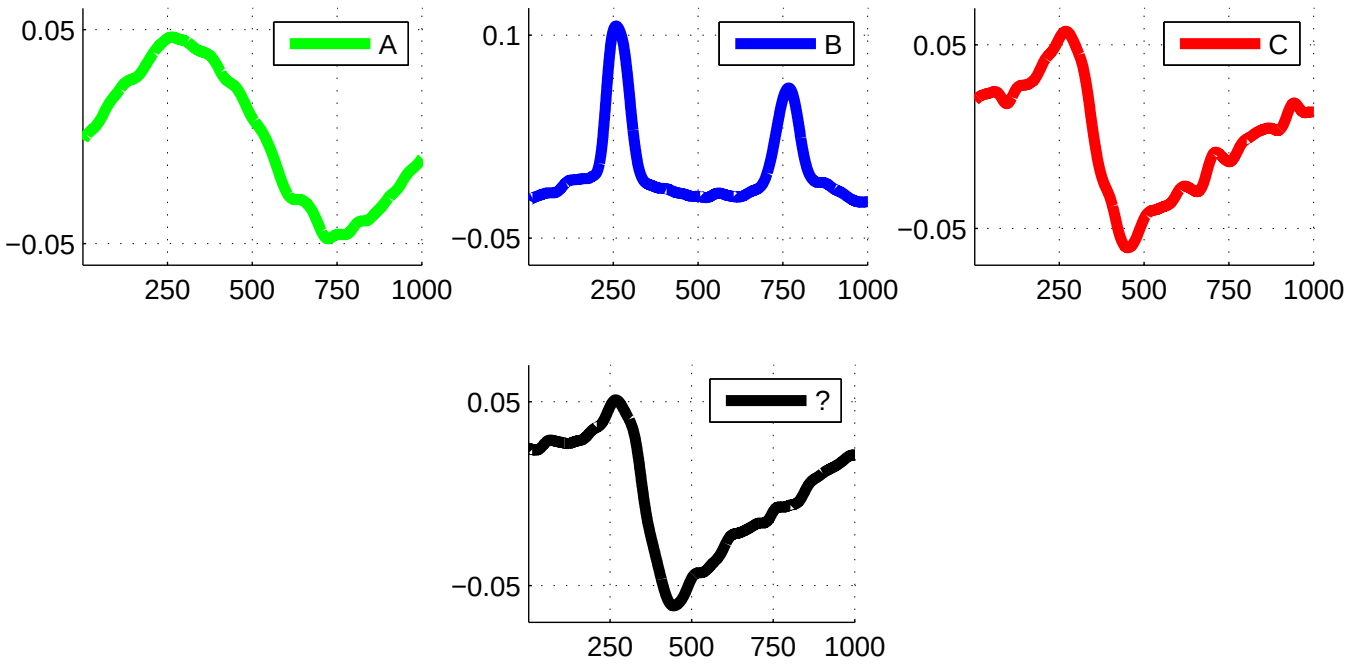
Time-series Classification

Dr. Josif Grabocka

ISMLL, University of Hildesheim

Business Analytics

Time-series classification



Labelled instances from the *StarLightCurves* dataset, with the objective of predicting the class of a new series (shown in black).

Problem Definition

- **Given** a set of training series $T^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times Q}$, and their expert-labeled targets $Y^{\text{Train}} \in \mathbb{N}^{N^{\text{Train}}}$,
- **Accurately predict** the unknown targets $Y^{\text{Test}} \in \mathbb{N}^{N^{\text{Test}}}$ of testing series $T^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times Q}$.

Problem Definition

- **Given** a set of training series $T^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times Q}$, and their expert-labeled targets $Y^{\text{Train}} \in \mathbb{N}^{N^{\text{Train}}}$,
- **Accurately predict** the unknown targets $Y^{\text{Test}} \in \mathbb{N}^{N^{\text{Test}}}$ of testing series $T^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times Q}$.
- Define a prediction model $\mathcal{M}$ having parameters $\theta \in \mathbb{R}^{D_{\text{P}}}$:

$$\hat{Y}_i := \mathcal{M}(T_i^*, \theta) : \left(\mathbb{R}^Q \times \mathbb{R}^{D_{\text{P}}}\right) \to \mathbb{R},\ i = 1, \ldots, N^*$$

Problem Definition

- **Given** a set of training series $T^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times Q}$, and their expert-labeled targets $Y^{\text{Train}} \in \mathbb{N}^{N^{\text{Train}}}$,
- **Accurately predict** the unknown targets $Y^{\text{Test}} \in \mathbb{N}^{N^{\text{Test}}}$ of testing series $T^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times Q}$.
- Define a prediction model $\mathcal{M}$ having parameters $\theta \in \mathbb{R}^{D_{\text{P}}}$:

$$\hat{Y}_i \;:=\; \mathcal{M}(T_i^*, \theta) \;:\; \left(\mathbb{R}^Q \times \mathbb{R}^{D_{\text{P}}}\right) \to \mathbb{R},\; i = 1, \ldots, N^*$$

- Find the parameters that minimize a regularized loss (e.g. MCR):

$$\theta_\gamma := \underset{\theta^* \in \mathbb{R}^{D_{\text{P}}}}{\operatorname{argmin}} \sum_{i=1}^{N^{\text{Train}}} \mathcal{L}(Y_i^{\text{Train}},\; \mathcal{M}(T_i^{\text{Train}}, \theta^*, \gamma)) \;+\; \mathcal{R}(\theta^*)$$

Problem Definition

- **Given** a set of training series $T^{\text{Train}} \in \mathbb{R}^{N^{\text{Train}} \times Q}$, and their expert-labeled targets $Y^{\text{Train}} \in \mathbb{N}^{N^{\text{Train}}}$,
- **Accurately predict** the unknown targets $Y^{\text{Test}} \in \mathbb{N}^{N^{\text{Test}}}$ of testing series $T^{\text{Test}} \in \mathbb{R}^{N^{\text{Test}} \times Q}$.
- Define a prediction model $\mathcal{M}$ having parameters $\theta \in \mathbb{R}^{D_{\text{P}}}$:

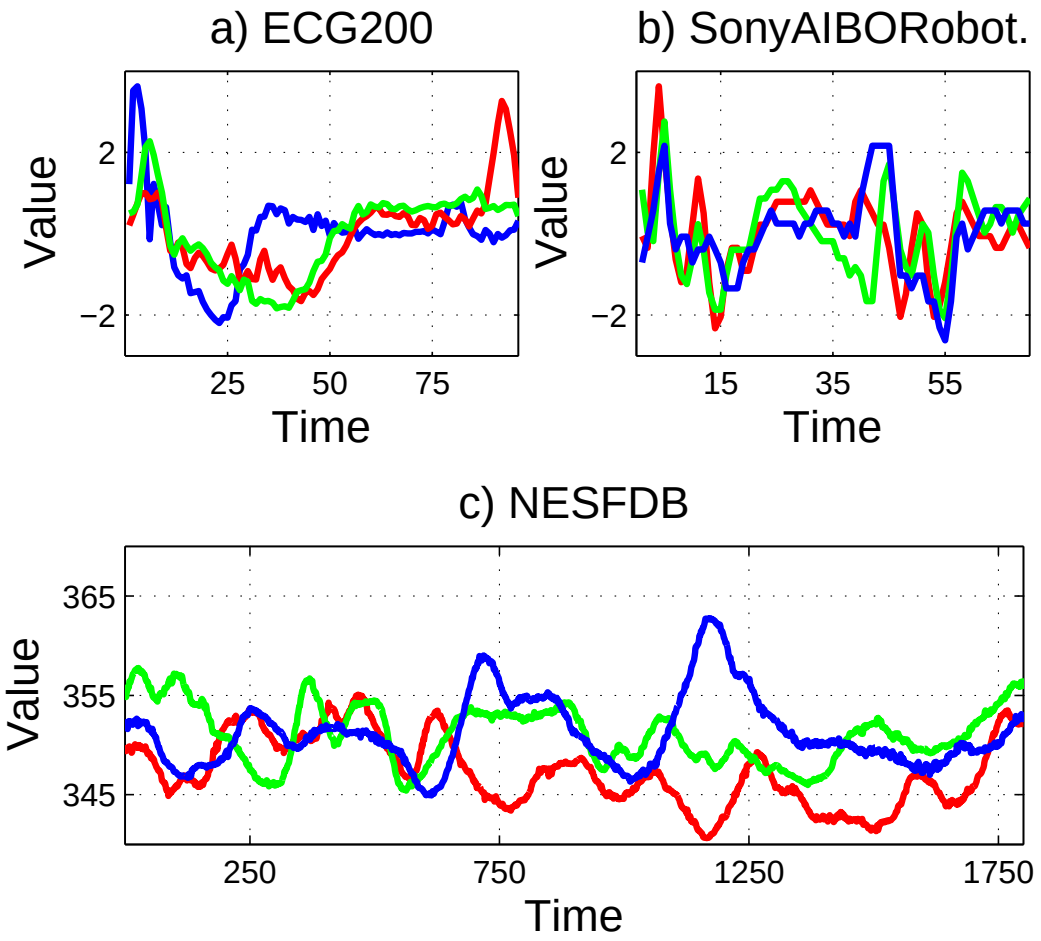
$$\hat{Y}_i \;:=\; \mathcal{M}(T_i^*, \theta) \;:\; \left(\mathbb{R}^Q \times \mathbb{R}^{D_{\text{P}}}\right) \to \mathbb{R},\; i = 1, \ldots, N^*$$

- Find the parameters that minimize a regularized loss (e.g. MCR):

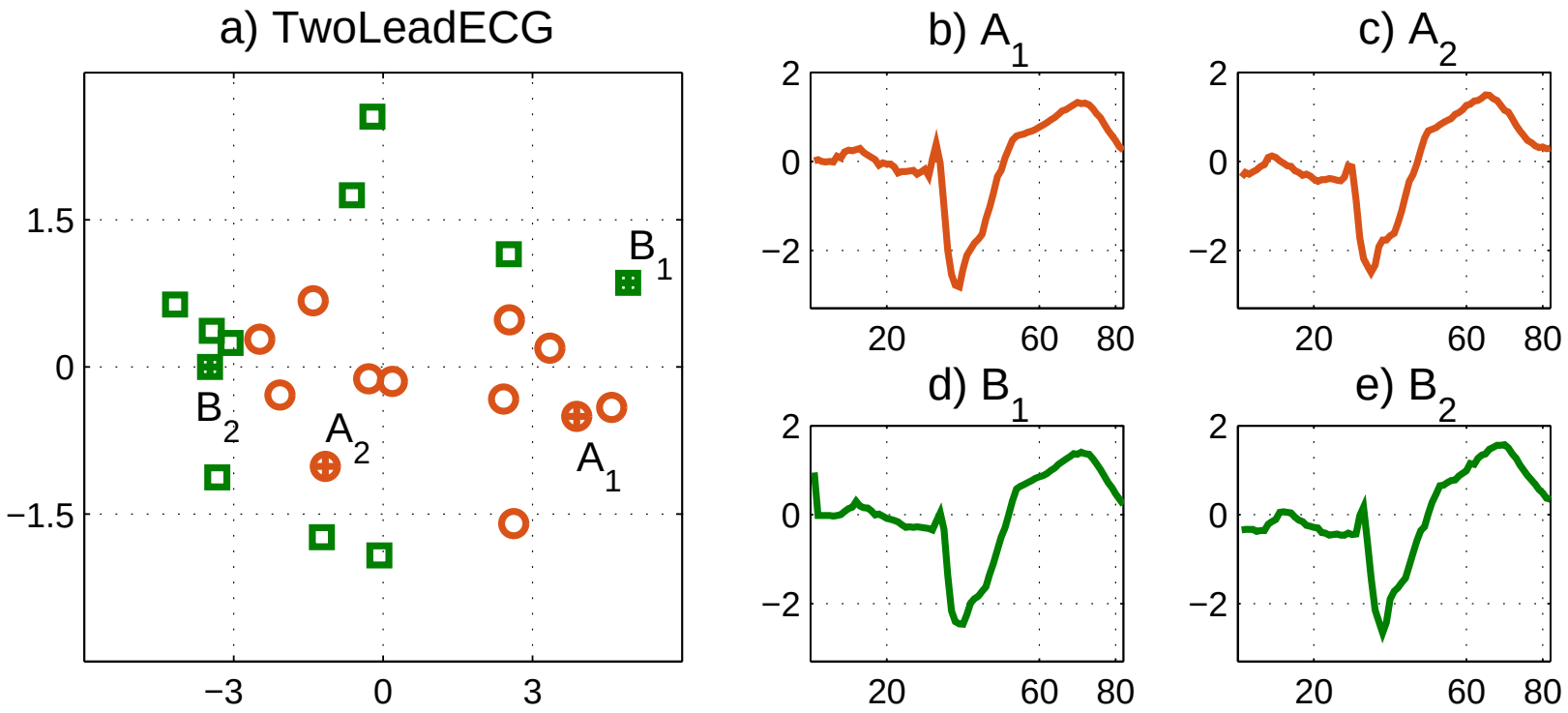
$$\theta_\gamma := \underset{\theta^* \in \mathbb{R}^{D_{\text{P}}}}{\operatorname{argmin}} \sum_{i=1}^{N^{\text{Train}}} \mathcal{L}(Y_i^{\text{Train}},\; \mathcal{M}(T_i^{\text{Train}}, \theta^*, \gamma)) \;+\; \mathcal{R}(\theta^*)$$

- Several models include hyper-parameters which need to be tuned.

Why is classifying time series nontrivial?



Intra-class variations



MDS of the Euclidean distances among series from the TwoLeadECG dataset

1-NN and Similarity Distance

The Nearest Neighbor classifier predicts the most similar target:

$$\begin{aligned} \hat{Y}_{\mathcal{T}} &\leftarrow Y_i^*, \text{ s.t.} \\ i^* &= \underset{j=1,\ldots,N^{\text{Train}}}{\text{argmin}} \quad \text{Sim}(T_j^{\text{Train}}, \mathcal{T}) \end{aligned}$$

The Euclidean distance:

$$\text{Sim}(T_j^{\text{Train}}, \mathcal{T}) := \text{ED}(T_j^{\text{Train}}, \mathcal{T}) = \sum_{q=1}^{Q} \left(T_{j,q}^{\text{Train}} - \mathcal{T}_q \right)^2$$

Dynamic Time Warping

Concretely, a warping path between two series $T_{i,:} = (T_{i,1}, ..., T_{i,Q})$ and $T_{j,:} = (T_{j,1}, ..., T_{j,Q})$, denoted as τ^{T_i,T_j} is defined as an alignment $\tau^{T_i,T_j} = (\tau_1^{T_i,T_j}, \tau_2^{T_i,T_j})$ between the indices of $T_{i,:}$ and $T_{j,:}$.

$$P = |\tau^{T_i,T_j}|$$
$$1 = \tau^{T_i,T_j}(1)_1 \leq ... \leq \tau^{T_i,T_j}(P)_1 = Q$$
$$1 = \tau^{T_i,T_j}(1)_2 \leq ... \leq \tau^{T_i,T_j}(P)_2 = Q$$

while involving incremental alignment of adjacent pairs as:

$$\begin{pmatrix} \tau^{T_i,T_j}(t+1)_1 - \tau^{T_i,T_j}(t)_1 \\ \tau^{T_i,T_j}(t+1)_2 - \tau^{T_i,T_j}(t)_2 \end{pmatrix} \in \left\{ \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\},$$
$$t = 1, \ldots, P.$$

Dynamic Time Warping (II)

$$\mathrm{DTW}(T_i, T_j) = \underset{\tau^{T_i,T_j}}{\operatorname{argmin}} \sum_{p=1}^{|\tau^{T_i,T_j}|} \left(T_{i,\tau^{T_i,T_j}(p)_1} - T_{j,\tau^{T_i,T_j}(p)_2} \right)^2$$

Practically computed by a dynamic algorithm in $\mathcal{O}(Q^2)$:

$$\begin{aligned} \mathrm{DTW}(T_i, T_j) &= \mathbf{W}_{\mathbf{Q},\mathbf{Q}}, \text{ such that:} \\ \mathbf{W}_{1,1} &= (T_{i,1} - T_{j,1})^2 \\ \mathbf{W}_{a,b} &= (T_{i,a} - T_{j,b})^2 + \min(\mathbf{W}_{a-1,b}, \mathbf{W}_{a,b-1}, \mathbf{W}_{a-1,b-1}) \\ & \quad a = 2, \ldots, Q, b = 2, \ldots, Q \end{aligned} \tag{1}$$

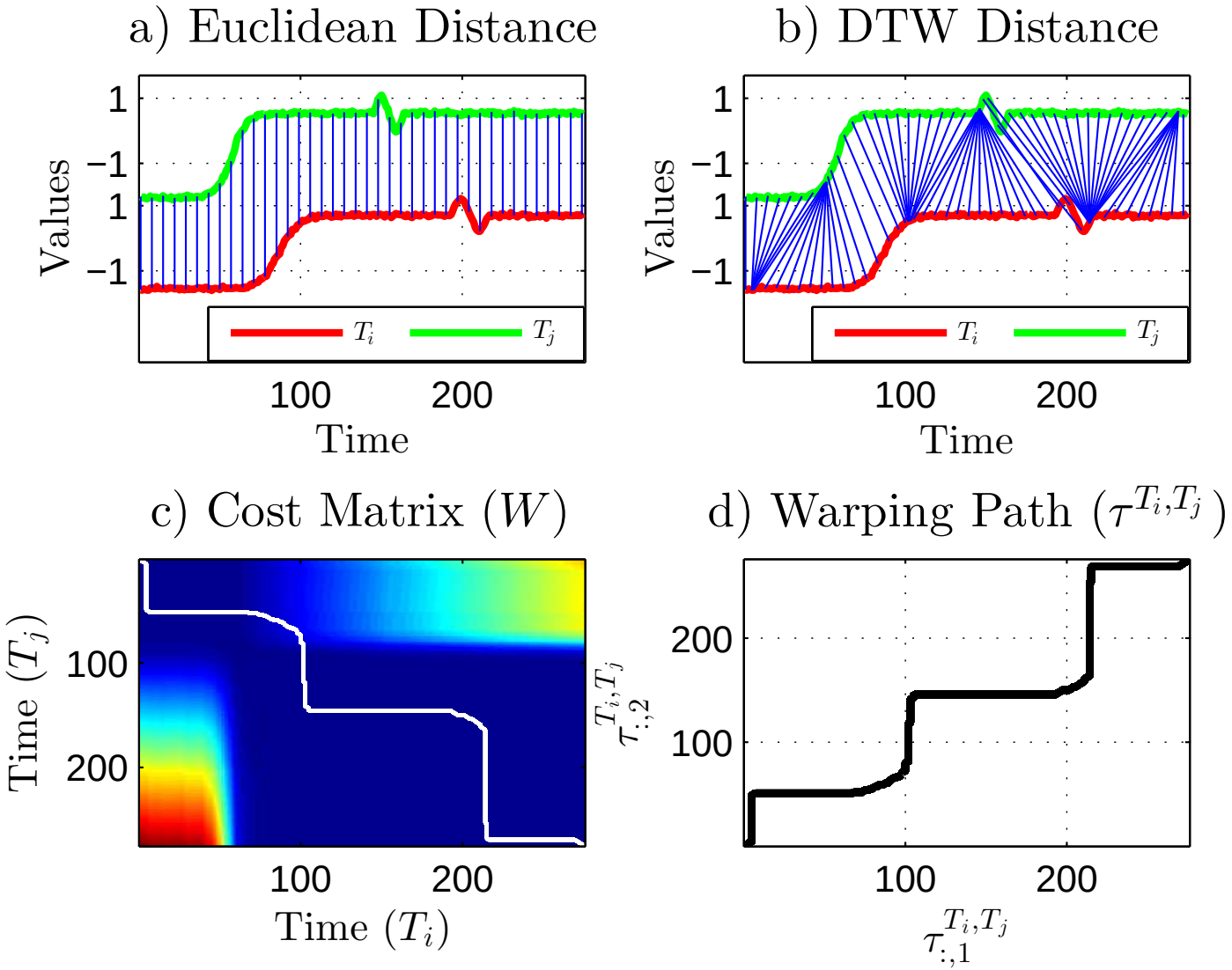
DTW Exercise

Given series $T_1 = (0, 1, 0, -1, 0)$ and $T_2 = (1, 0, -1, 0, 0)$

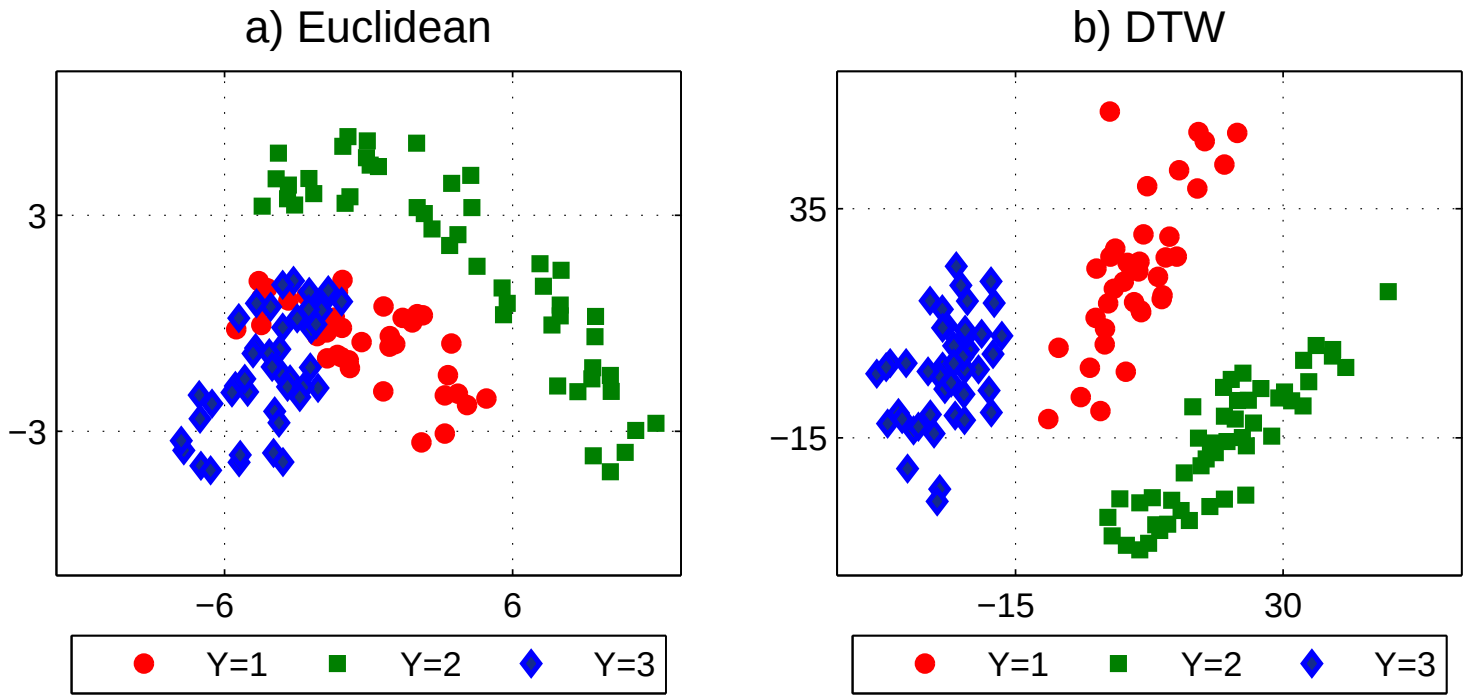
a) Compute the Euclidean distance: $\text{ED}(T_1, T_2) = ?$

b) Compute their DTW distance: $\text{DTW}(T_1, T_2) = ?$

Illustration: Euclidean vs DTW

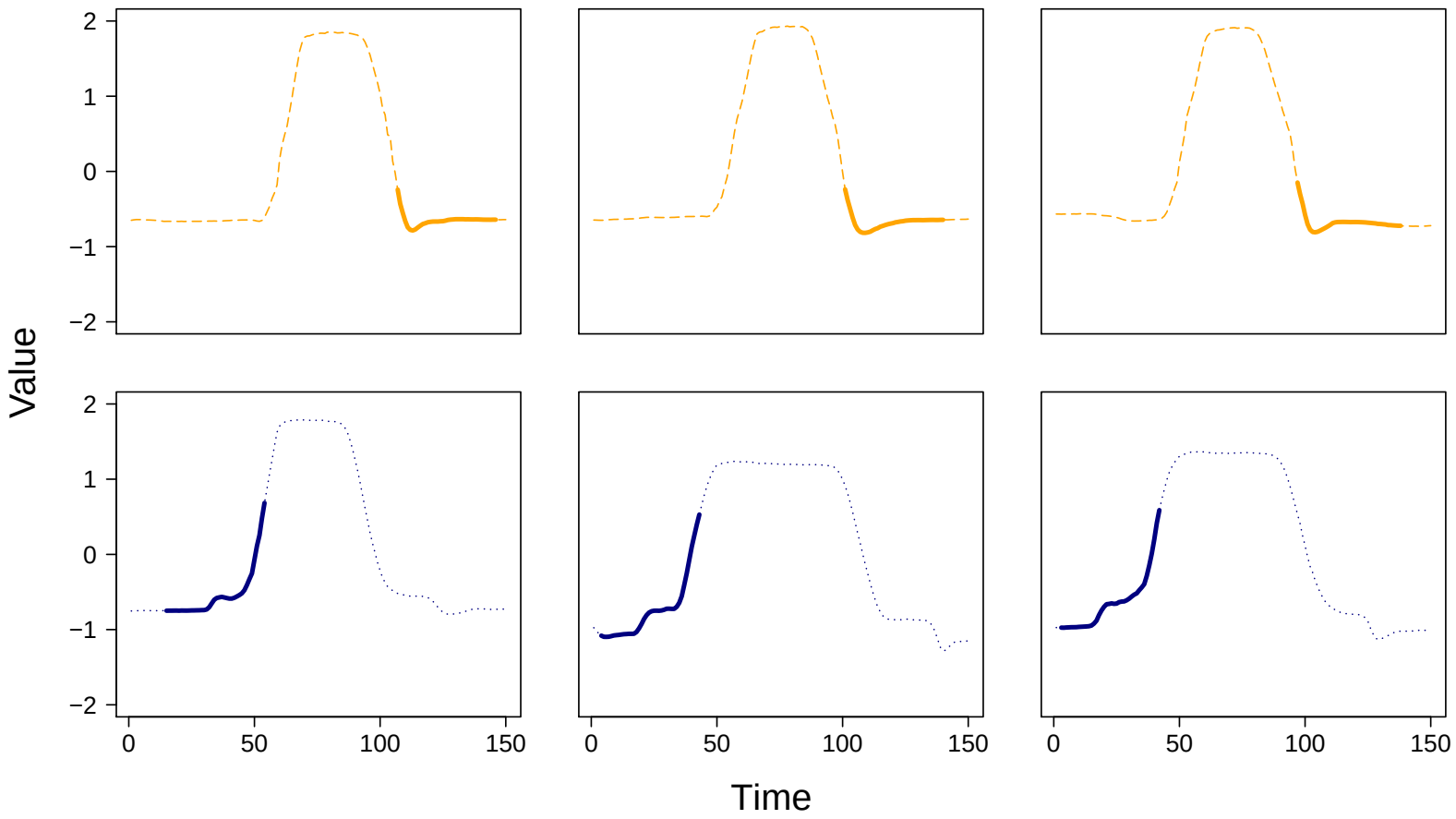


Effect of Euclidean and DTW on 1-NN



Multi-dimensional scaling: Euclidean vs. Dynamic Time Warping

Discriminative Subsequences

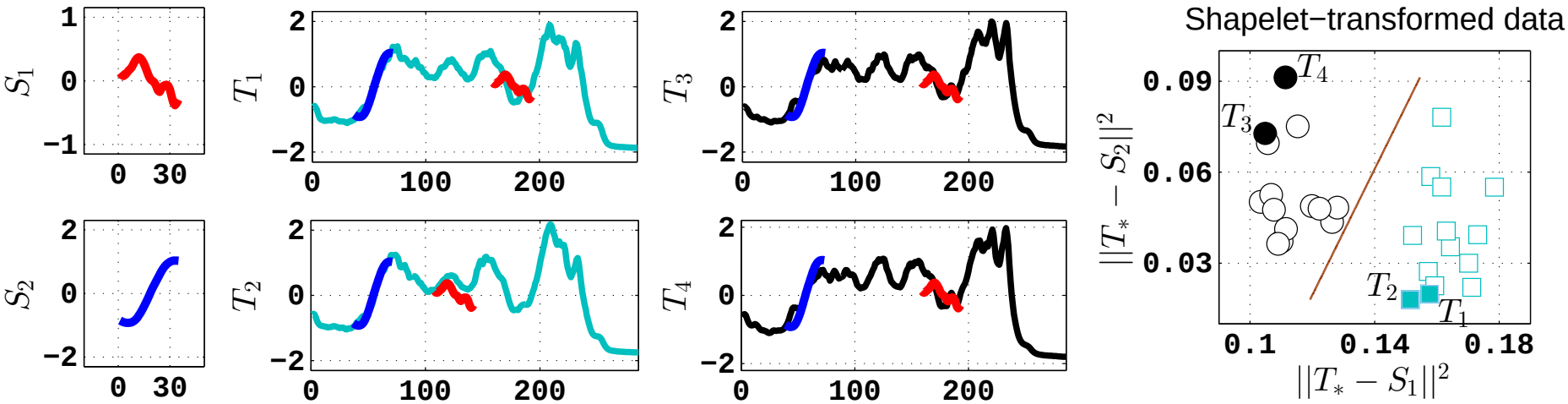


Time-Series Shapelets

Minimum distances $M \in \mathbb{R}^{N \times K}$ between shapelets $S \in \mathbb{R}^{K \times L}$ and all J segments of series $T \in \mathbb{R}^{N \times Q}$:

$$M_{i,k} = \min_{j=1,\ldots,J} \frac{1}{L} \sum_{l=1}^{L} (T_{i,j+l-1} - S_{k,l})^2$$

... yield discriminative predictors:



Left: Shapelets, Middle: Match, Right: Shapelet-Transformation

Prediction Model and Loss Function

A linear model of predictors $M \in \mathbb{R}^{N \times K}$ and weights $W \in \mathbb{R}^K, W_0 \in \mathbb{R}$: can be used to estimate the target $\hat{Y} \in \mathbb{R}^N$:

$$\hat{Y}_i \quad = \quad W_0 + \sum_{k=1}^{K} M_{i,k} W_k \qquad \forall i \in \{1, \ldots, N\}$$

The risk of estimating the true target $Y \in \{0, 1\}^N$ from approximated target $\hat{Y} \in \mathbb{R}^N$ is the logistic loss $\mathcal{L}(Y, \hat{Y}) \in \mathbb{R}^N$:

$$\mathcal{L}(Y_i, \hat{Y}_i) = -Y_i \ln \sigma(\hat{Y}_i) - (1 - Y_i) \ln \left(1 - \sigma(\hat{Y}_i)\right), \quad \forall i \in \{1, \ldots, N\}$$

Objective Function

The objective function $\mathcal{F} \in \mathbb{R}$ is a regularized loss function:

$$\underset{S,W}{\text{argmin}}\ \mathcal{F}(S,W) = \underset{S,W}{\text{argmin}} \sum_{i=1}^{N} \mathcal{L}(Y_i, \hat{Y}_i) + \lambda_W ||W||^2$$

The objective function $\mathcal{F}$ can be decomposed into per-instance objectives $\mathcal{F}_i$:

$$\mathcal{F}_i \quad = \quad \mathcal{L}(Y_i, \hat{Y}_i) + \frac{\lambda_W}{N} \sum_{k=1}^{K} {W_k}^2, \qquad \forall i \in \{1, \ldots, N\}$$

Mission: Learn S, W that minimize $\mathcal{F}$.

Differentiable Minimum Function

Approximate the true minimum M with the soft-minimum version $\hat{M}$:

$$M_{i,k} \approx \hat{M}_{i,k} = \frac{\sum_{j=1}^{J} D_{i,k,j}\ e^{\alpha\, D_{i,k,j}}}{\sum_{j'=1}^{J} e^{\alpha\, D_{i,k,j'}}},$$
$$\alpha \in (-\infty, 0] \quad \forall i \in \{1,\ldots,N\}, \forall k \in \{1,\ldots,K\}$$

$$D_{i,k,j} := \frac{1}{L}\sum_{l=1}^{L}\left(T_{i,j+l-1} - S_{k,l}\right)^2,$$
$$\forall i \in \{1,\ldots,N\}, \forall k \in \{1,\ldots,K\}, \forall j \in \{1,\ldots,J\}$$

Partial Gradients of Objective

The partial derivative of the per-instance objective function $\mathcal{F}_i$ with respect to the l-th point of the k-th shapelet $S_{k,l}$ is computed using the chain rule of derivation:

$$\frac{\partial \mathcal{F}_i}{\partial S_{k,l}} = \frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} \sum_{j=1}^{J} \frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} \frac{\partial D_{i,k,j}}{\partial S_{k,l}} \tag{2}$$

$$\frac{\partial \mathcal{F}_i}{\partial W_k} = \frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} \frac{\partial \hat{Y}_i}{\partial \hat{W}_k} + \frac{\partial \mathrm{Reg}(W)}{\partial W_k}, \quad \frac{\partial \mathcal{F}_i}{\partial W_0} = \frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} \tag{3}$$

All the components of the partial derivative are computable as follows:

$$\frac{\partial \mathcal{L}(Y_i, \hat{Y}_i)}{\partial \hat{Y}_i} = -\left(Y_i - \sigma\left(\hat{Y}_i\right)\right), \quad \frac{\partial \hat{Y}_i}{\partial \hat{M}_{i,k}} = W_k, \quad \frac{\partial \hat{Y}_i}{\partial \hat{W}_k} = M_{i,k}, \quad \frac{\partial \mathrm{Reg}(W)}{\partial W_k} = \frac{2\lambda_W}{I} W_k \tag{4}$$

$$\frac{\partial \hat{M}_{i,k}}{\partial D_{i,k,j}} = \frac{e^{\alpha D_{i,k,j}} \left(1 + \alpha \left(D_{i,k,j} - \hat{M}_{i,k}\right)\right)}{\sum_{j'=1}^{J} e^{\alpha D_{i,k,j'}}}, \frac{\partial D_{i,k,j}}{\partial S_{k,l}} = \frac{2}{L}\left(S_{k,l} - T_{i,j+l-1}\right) \tag{5}$$

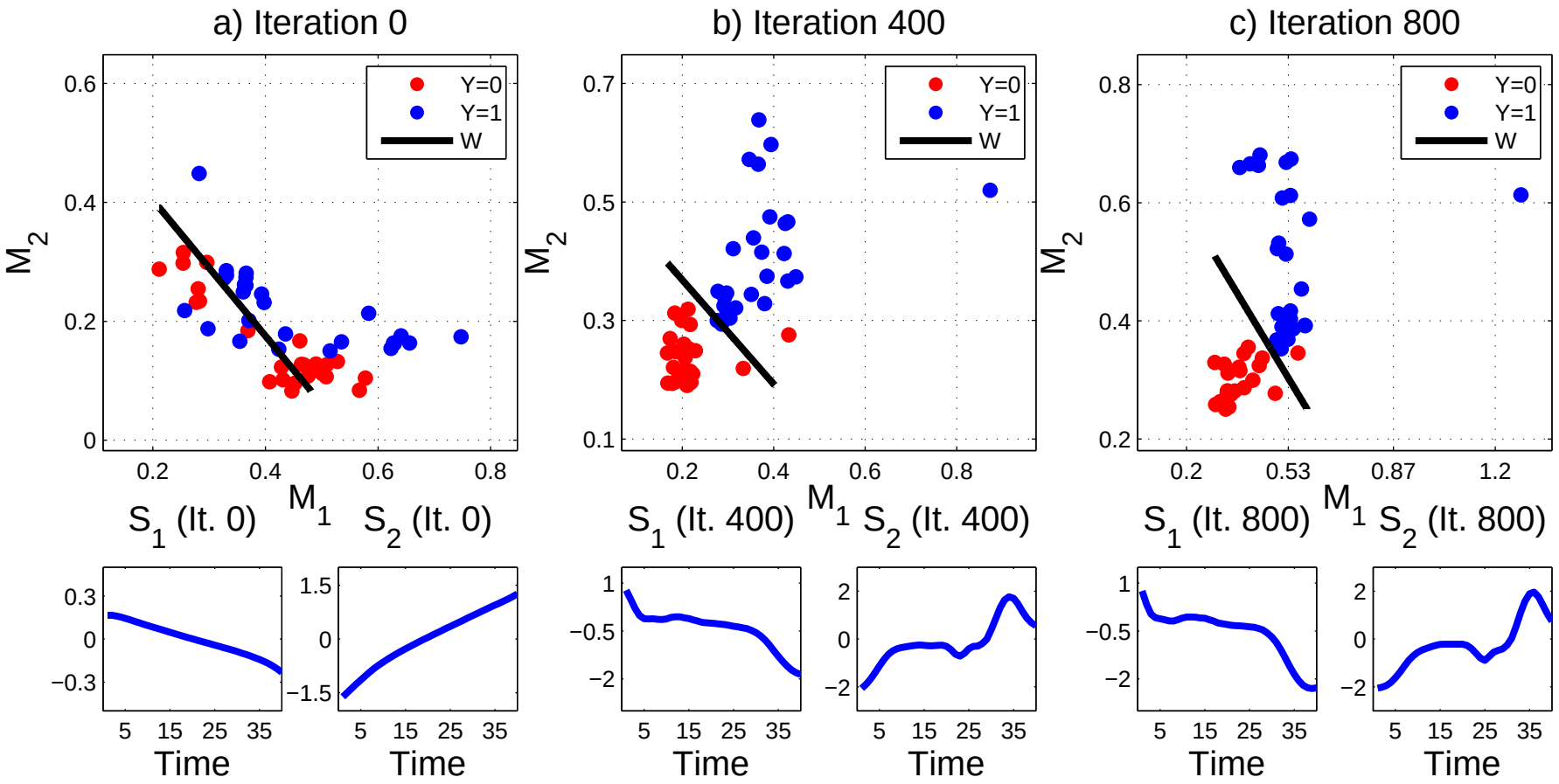
Learning Time-series Shapelets

Require: $T \in \mathbb{R}^{N \times Q}$, Number of Shapelets K, Length of a shapelet L, Regularization λ_W, Learning Rate η, Number of iterations: maxIter

Ensure: Shapelets $S \in \mathbb{R}^{K \times L}$, Classification weights $W \in \mathbb{R}^K$, Bias $W_0 \in \mathbb{R}$

for iteration=$1, \ldots, maxIter$ **do**
  for $i = 1, \ldots, N$ **do**
    $W_0 \leftarrow W_0 - \eta \frac{\partial \mathcal{F}_i}{\partial W_0}$
    for $k = 1, \ldots, K$ **do**
      $W_k \leftarrow W_k - \eta \frac{\partial \mathcal{F}_i}{\partial W_k}$
      for $L = 1, \ldots, L$ **do**
        $S_{k,l} \leftarrow S_{k,l} - \eta \frac{\partial \mathcal{F}_i}{\partial S_{k,l}}$
return S, W, W_0

Illustration of Shapelets Learning



Learning Two Shapelets on the Gun-Point Dataset

Time-series Representation - PAA and SAX

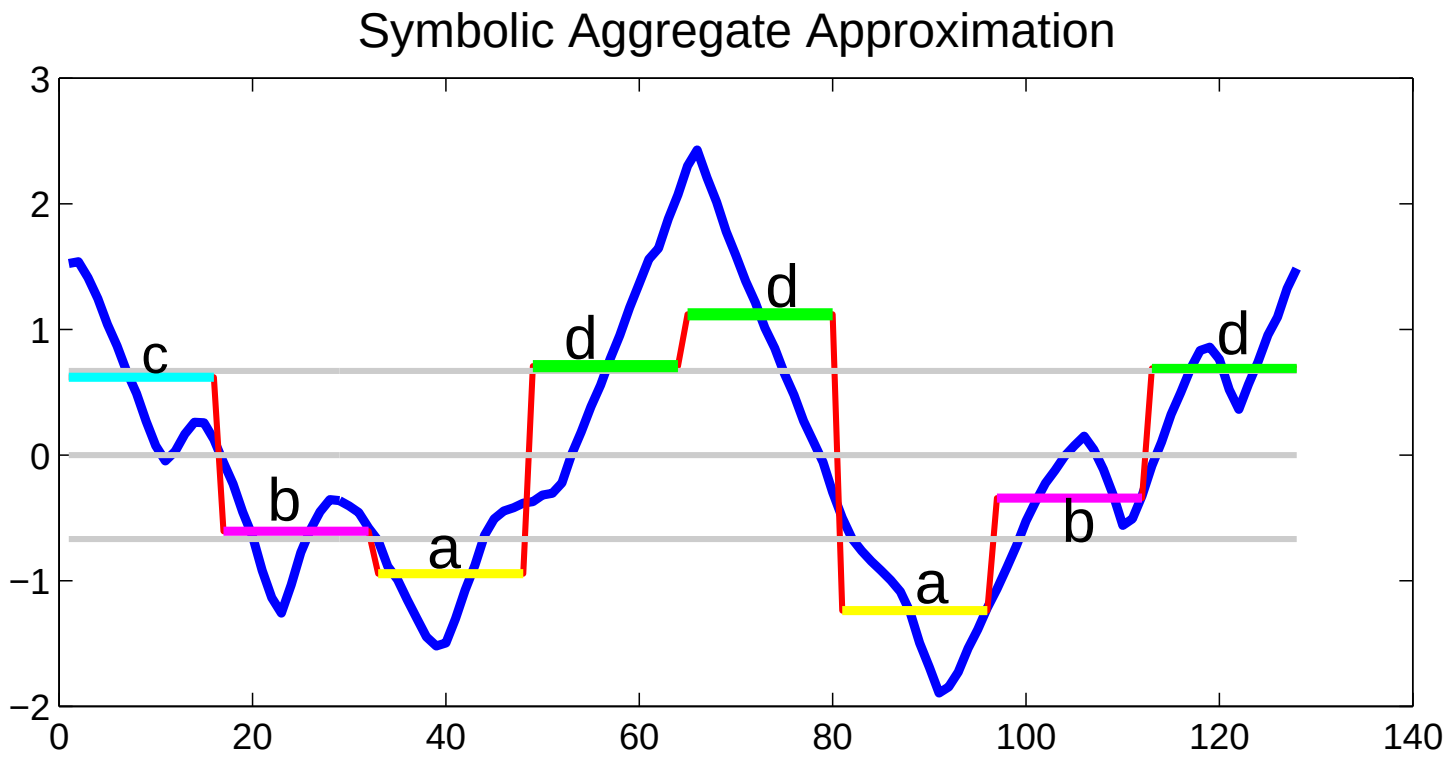
Piecewise Aggregate Approximation (PAA): Replace non-overlapping segments of length L, by its average:

$$T_{i,m}^{\text{PAA}} := \frac{1}{L}\sum_{l=1}^{L} T_{i,l+(m-1)L}, \quad i = 1,\ldots,N;\ m = 1,\ldots,\frac{Q}{L}$$

Symbolic Aggregate Approximation (SAX): Assign the PAA values into a symbol from an alphabet of w symbols $(s_1, s_2, \ldots, s_w)$, by assigning start ψ_{s_v} and end values ϕ_{s_v} for a symbol to each symbol s_v:

$$\begin{aligned} T_{i,m}^{\text{SAX}} \quad := \quad & s_v \text{ if } \psi_{s_v} \leq T_{i,m}^{\text{PAA}} < \phi_{s_v}; \\ & i = 1,\ldots,N;\ m = 1,\ldots,\frac{Q}{L}, v \in \{1,\ldots,w\} \end{aligned}$$

SAX Illustration



- Overlapping symbols of length n, create a series of words per series
- The series above has words ($n = 2$): cb, ba, ad, dd, da, ab, bd

Bag of SAX Words

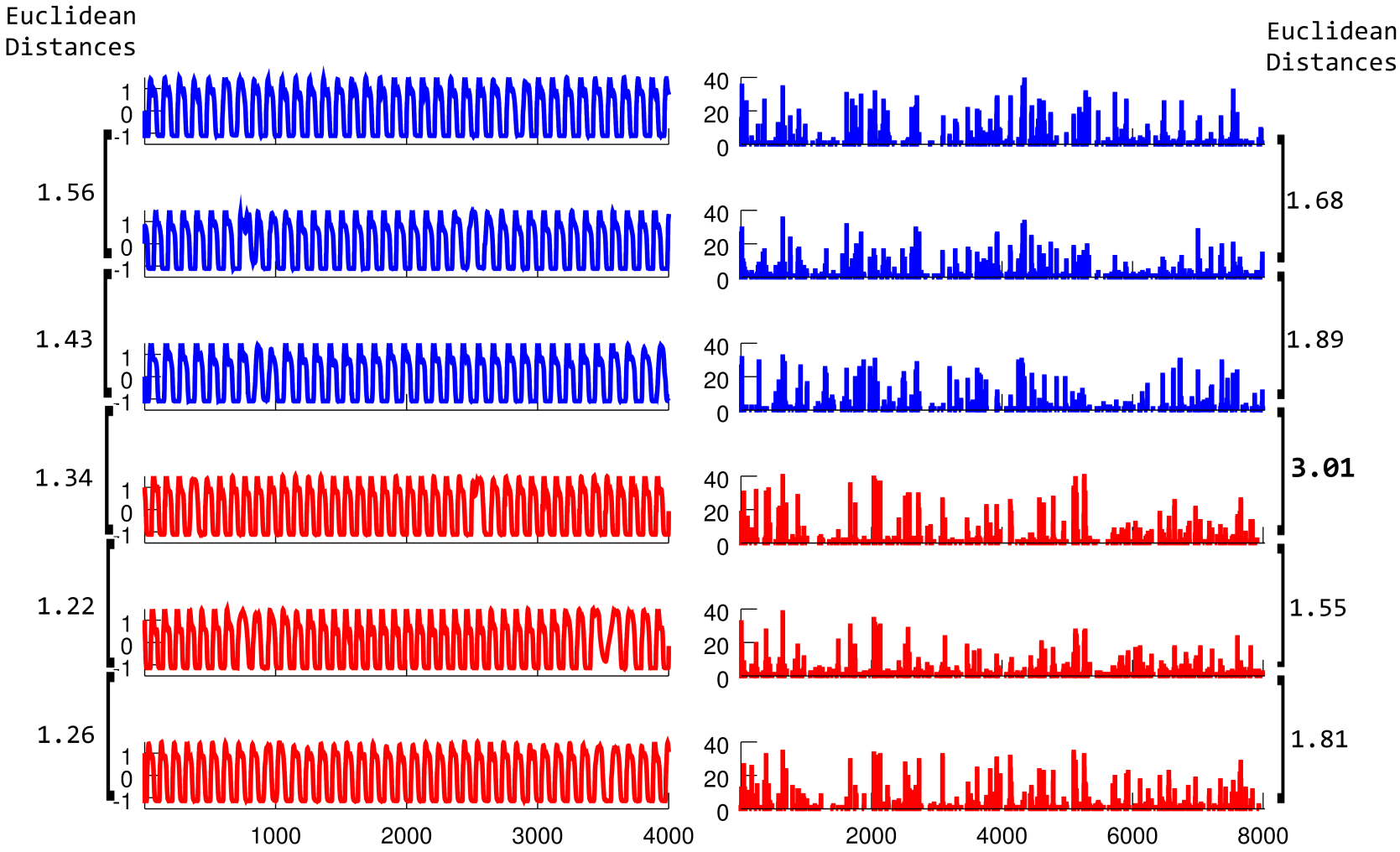
Let the words of the i-th series be $S^{\text{SAX}}_{i,1}, \ldots, S^{\text{SAX}}_{i,M_i}$. The dictionary of all words:

$$\mathcal{D} := \bigcup_{i=1}^{N} \bigcup_{j=1}^{M_i} S^{\text{SAX}}_{i,j}$$

The frequency of words occurring in a series define a bag-of-words representation:

$$T^{\text{BoW}}_{i,z} := \left| \left\{ j \in \{1, \ldots, M_i\} \mid S^{\text{SAX}}_{i,j} = \mathcal{D}_z \right\} \right|; \quad i = 1, \ldots, N, z = 1, \ldots, |\mathcal{D}|$$

Advantage of Bag-of-words methods



Original time series vs. histogram of local patterns