Chapter 4

Multi-modal and Multi-scale Time series Metric Learning (M^2TML) implementation

Sommaire

In this chapter, we present the steps of our proposed algorithm referred as Multi-modal and Multi-scale Time series Metric Learning (M^2TML).
First, we introduce the multi-scale comparison concept for time series. Then, we present the steps to learn a Multi-modal and Multi-scale Time series metric for a robust k-Nearest Neighbor classifier: projection in the pairwise space, neighborhood construction and scaling, SVM-based metric learning resolution, and definition of the dissimilarity measure. We conclude by extending the algorithm M^2TML for multivariate and regression problems.

4.1 Multi-scale approach

In some applications, time series may exhibit similarities among the classes based on local patterns in the signal. Fig. 4.1 illustrates a toy example from the BME dataset in which time series of different classes seems to be similar on a global scale. However, at a more locally scale, a characteristic upward bell at the beginning or at the end of the time series allows to differentiate the class B (upward bell at the beginning) from the class E (upward bell at the end). Also, in massive time series datasets, computing the metric on all time series elements x_{it} might become time consuming. Computing the metric on a smaller part of the signal and not all the time series elements x_{it} makes the metric computation faster.

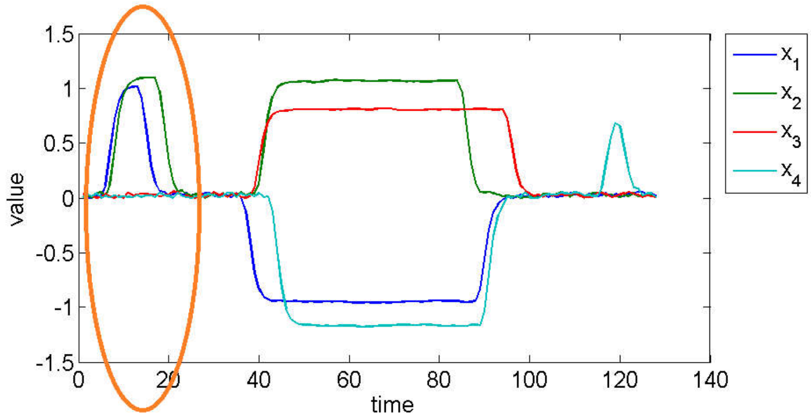

Figure 4.1: Example of 4 time series from the BME dataset, made of 3 classes : Begin, Middle and End. The 'Up' class has a characteristic bell at the beginning of the time series. The 'End' class has a characteristic bell at the end of the time series. The 'Middle' class has no characteristic bell. Orange circle show the region of interest of these bells for the class 'Begin'. This region is local and standard global metric fails to show these characteristics.

Localizing patterns of interest in huge time series datasets has become an active area of search in many applications including diagnosis and monitoring of complex systems, biomedical data analysis, and data analysis in scientific and business time series . A large number of methods have been proposed covering the extraction of local features from temporal windows [BC94] or the matching of queries according to a reference sequence [FRM94]. We focus on the computation of "local metrics".

It can be noted that the distance measures (amplitude-based d_A[1], frequential-based d_F, behavior-based d_B) in Eqs. 2.1, 2.4 and 2.6 implies systematically the total time series elements x_{it} and thus, restricts the distance measures to capture local temporal differences. In our work, we provide a multi-scale framework for time series comparison using a hierarchical structure. Many methods exist in the literature such as the sliding window or the dichotomy . We detailed here the latter one.

A multi-scale description can be obtained by repeatedly segmenting a time series expressed at a given temporal scale to induce its description at a more locally level. Many approaches have been proposed assuming fixed either the number of the segments or their lengths. In our work, we consider a binary segmentation at each level. Let $I = [a; b]$ be a temporal interval of size $(b - a)$. The interval I is decomposed into two equal overlaped intervals I_L (left interval) and I_R (right interval). A parameter α that allows to overlap the two intervals I_L and I_R,

[1] We recall that d_A is the Euclidean distance d_E in our work.

covering discriminating subsequences in the central region of I (around $\frac{b-a}{2}$):

$$I = [a; b] \tag{4.1}$$

$$I_L = [a; a + \alpha(b - a)] \tag{4.2}$$

$$I_R = [a - \alpha(b - a); b] \tag{4.3}$$

For $\alpha = 0.6$, the overlap covers 10% of the size of the interval I. Then, the process is repeated on the intervals I_L and I_R. We obtain a set of intervals I_s illustrated in Fig. 4.2.

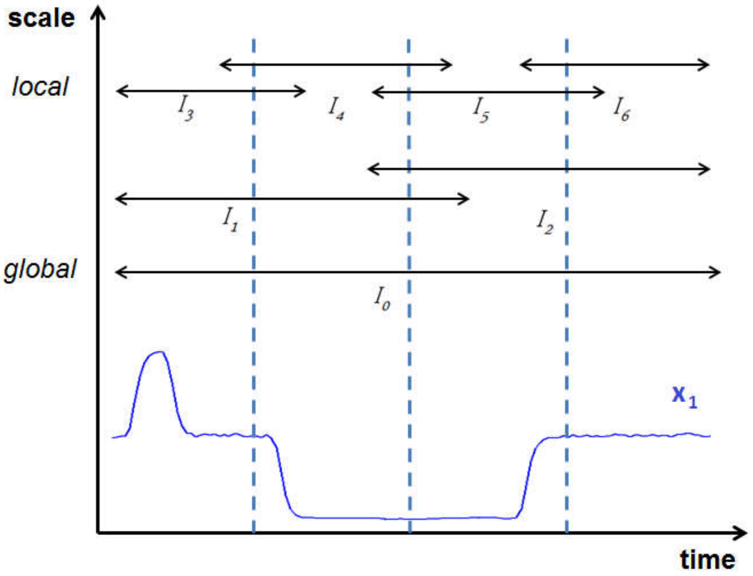

Figure 4.2: Multi-scale decomposition

A multi-scale description is obtained on computing the usual time series metrics (d_A, d_B, d_F) on the resulting segments I_s. Note that for two time series $\mathbf{x}_i$ and $\mathbf{x}_j$, the comparison between $\mathbf{x}_i$ and $\mathbf{x}_j$ is done on the same interval I_s. For a multi-scale amplitude-based comparison based on binary segmentation, the set of involved amplitude-based measures d_A^{Is} is:

$$d_A^{Is}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t \in Is} (x_{it} - x_{jt})^2} \tag{4.4}$$

The local behaviors- and frequential- based measures d_B^{Is} and d_F^{Is} are obtained similarly.

4.2 Projection in the pairwise space

Let $\{\mathbf{x}_i; y_i\}_{i=1}^n$ be n time series $\mathbf{x}_i \in \mathbb{R}^Q$ of length Q and of class label y_i. Let $d_1, \ldots, d_p$ be the set of multi-modal and multi-scale dissimilarity measures $d_A^{I_s}$, $d_B^{I_s}$ and $d_F^{I_s}$ related to segments I_s of several temporal scales, as described in Section 4.1.

Projection in the pairwise space
We note ψ an embedding function that maps each pair of time series $(\mathbf{x}_i; \mathbf{x}_j)$ to a vector $\mathbf{x}_{ij}$ in a dissimilarity space $\mathbb{R}^p$ whose dimensions are the dissimilarities $d_1, \dots, d_p$ as explained in Chapter 3:

$$\begin{aligned} \psi : \mathbb{R}^Q \times \mathbb{R}^Q &\to \mathscr{E} \\ (\mathbf{x}_i; \mathbf{x}_j) &\to \mathbf{x}_{ij} = [d_1(\mathbf{x}_i; \mathbf{x}_j), \dots, d_p(\mathbf{x}_i; \mathbf{x}_j)]^T \end{aligned} \tag{4.5}$$

We cast the problem of learning a multi-modal and multiscale temporal metric as learning the metric D a combination function of $d_1, \dots, d_p$ in the pairwise space $\mathscr{E}$:

$$D = f(d_1, \dots, d_p) \tag{4.6}$$

The learning process is guided by local constraints to ensure dissimilarities between neighbors of a same class (i.e. $y_{ij} = -1$) lower than the dissimilarity between neighbors of different classes ($y_{ij} = +1$). The learned metric D should satisfy, in addition, the properties of a dissimilarity measure, i.e. positivity ($D(\mathbf{x}_{ij}) \geq 0$), distinguishability ($D(\mathbf{x}_{ij}) = 0, \mathbf{x}_i = \mathbf{x}_j$) and symmetry ($D(\mathbf{x}_{ij}) = D(\mathbf{x}_{ji})$).

Pairwise space normalization
The scale between the p basic metrics d_h can be different. Thus, there is a need to scale the data within the pairwise space and ensure comparable ranges for the p basic metrics d_h. In our experiment, we use dissimilarity measures with values in $[0; +\infty[$. Therefore, we propose to Z-normalize their log distributions.

4.3 Neighborhood construction and scaling

The metric learning problem aims to learn a metric D that pulls the k nearest neighbors (targets) while pushing the time series of different classes. Thus, the preliminary step defines the target pairs. For that, an initial distance is necessary to build the neighborhood.

Let $\mathbf{x}_{ij} \in \mathbb{R}^p$ $i, j \in \{1, \dots, n\}$ be a set of samples into the pairwise space $\mathscr{E}$ as described in Eq. 4.5. For each time series $\mathbf{x}_i$, we denote X_i^+ the set of **positive pairs** $\mathbf{x}_{ij}$ such that $y_{ij} = +1$ (i.e. the time series $\mathbf{x}_i$ and $\mathbf{x}_j$ has different class label $y_j \neq y_i$). Similarly, we denote X_i^- the set of **negative pairs** $\mathbf{x}_{ij}$ such that $y_{ij} = -1$ (i.e. the time series $\mathbf{x}_i$ and $\mathbf{x}_j$ has the same class label $y_j = y_i$):

$$X_i^- = \{\mathbf{x}_{ij}, y_{ij} = -1\} \quad \text{(same class)} \tag{4.7}$$

$$X_i^+ = \{\mathbf{x}_{ij}, y_{ij} = +1\} \quad \text{(different classes)} \tag{4.8}$$

As the learned distance D is not known, without prior knowledge, we choose a L_2 norm as an initial metric to define the positive and negative sets:

$$||\mathbf{x}_{ij}||_2 = \sqrt{\sum_{h=1}^{p} (d_h(\mathbf{x}_i, \mathbf{x}_j))^2} \tag{4.9}$$

The **target set** X_i^{-*} is a subset of the negative set X_i^- of pairs $\mathbf{x}_{ij}$ such that the time series $\mathbf{x}_j$ are the k-nearest neighbors of $\mathbf{x}_i$, denoted $j \rightsquigarrow i$:

$$X_i^{-*} = \{\mathbf{x}_{ij}, y_{ij} = -1\} \quad \text{s.t. } j \rightsquigarrow i \tag{4.10}$$

The k nearest neighbors of a sample $\mathbf{x}_i$, denoted $\mathbf{x}_j$ $(j \rightsquigarrow i)$, are defined in the pairwise space $\mathscr{E}$ by the k-th lowest norm $||\mathbf{x}_{ij}||_2$ negative pairs. Similarly, the **imposter set** X_i^{+*} is a subset of the positive set X_i^+ of pairs $\mathbf{x}_{il}$ such that the time series $\mathbf{x}_l$ is an imposter of $\mathbf{x}_i$, denoted $l \nrightarrow i$. It corresponds the pairs $\mathbf{x}_{il}$ that have a L_2 norm lower that the L_2 norm of the k-th nearest neighbor:

$$X_i^{+*} = \{\mathbf{x}_{il}, y_{il} = +1\} \quad \text{s.t. } l \nrightarrow i \tag{4.11}$$

To build the pairwise training set X_p, three solutions are proposed, illustrated in Fig 4.3:

1. **k-NN vs impostors**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and impostor set:
$$X_p = \bigcup_i (X_i^{-*} \cup X_i^{+*}) \tag{4.12}$$

2. **k-NN vs all**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and positive set. It ensures that no pairs $\mathbf{x}_{il}$ of different classes will invade the target neighborhood during the learning process:
$$X_p = \bigcup_i (X_i^{-*} \cup X_i^{+}) \tag{4.13}$$

3. **m-NN$^+$ vs m-NN$^-$**: it corresponds to the union for all $\mathbf{x}_i$ of the set of the m-nearest neighbors of the same class, denoted m-NN$_i^-$, and the m-nearest neighbor of $\mathbf{x}_i$ of a different class $(y_j \neq y_i)$, denoted m-NN$_i^+$. For a k-NN classifier, by considering larger neighborhoods with $m = \alpha k (\alpha > 1)$ one includes more variability to generalize better the obtained solution:
$$X_p = \bigcup_i (m\text{-NN}_i^+ \cup m\text{-NN}_i^-) \tag{4.14}$$

   In the following, for simplification purpose, we define m-NN$^- = \bigcup_i m$-NN$_i^-$ as the union of all of the set of the m-nearest neighbors of the same class and m-NN$^- = \bigcup_i m$-NN$_i^+$ as the m-nearest neighbor of $\mathbf{x}_i$ of a different class.

In our experiment, we use the m-NN$^+$ vs m-NN$^-$ strategy for better generalization of the solution compared to k-NN vs impostors strategy and for faster solutions compared to k-NN vs all strategy. Note that in m-NN$^+$ vs m-NN$^-$ strategy, the set of positive and negative pairs is balanced.

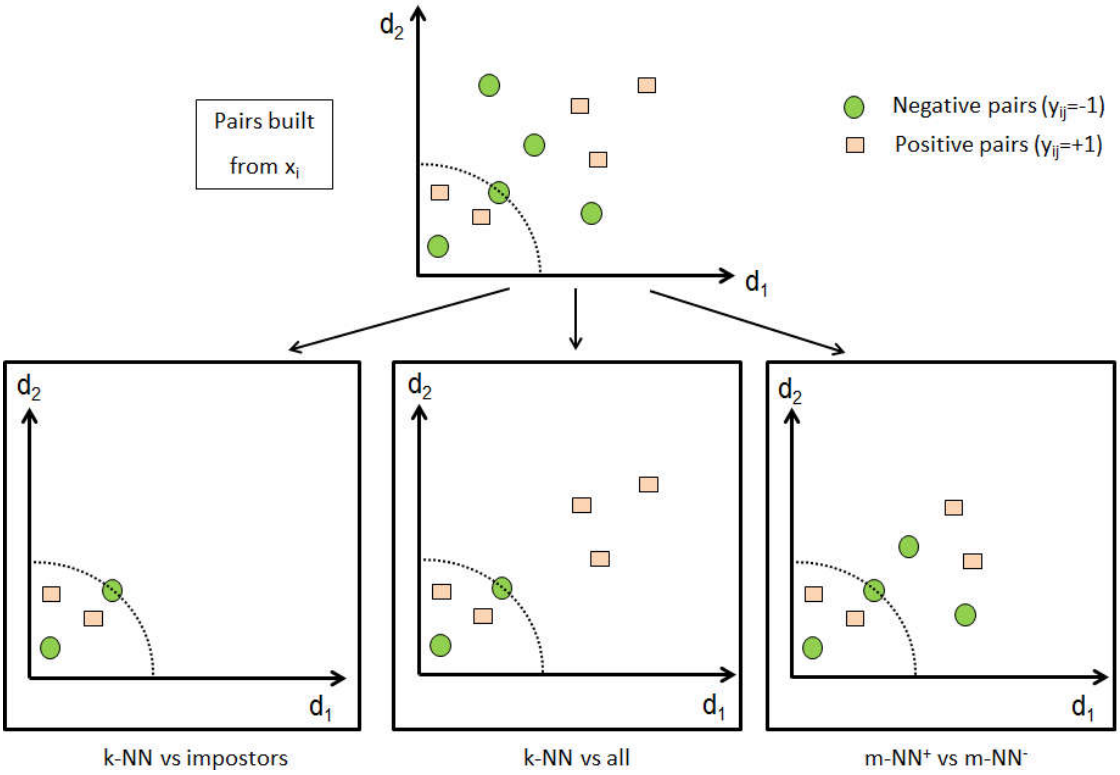

Figure 4.3: Example of a k-NN problem with $k = 2$. 3 different strategies (bottom) for pairwise training set X_p construction from the embedding of time series $\mathbf{x}_i$ in the pairwise space (top): k-NN vs impostor strategy (left), k-NN vs all strategy (middle) and m-NN$^+$ vs m-NN$^-$ (right) with $m = 4$.

Neighborhood scaling

Let r_i be the radius associated to $\mathbf{x}_i$ corresponding to the maximum norm of its m-th nearest neighbor of same class in m-NN$^-$:

$$r_i = \max_{\mathbf{x}_{ij} \in m\text{-NN}^-} ||\mathbf{x}_{ij}||_2 \tag{4.15}$$

As explained in Chapter 3, Section 3.4.3, there exists an heterogeneity in the neighborhood. In real datasets, local neighborhoods can have very different scales as illustrated in Fig. 3.8. To make the target neighborhood spreads comparable, we propose for each $\mathbf{x}_i$ to scale its neighborhood vectors $\mathbf{x}_{ij}$ such that the L_2 norm (radius) of the farthest m-th nearest neighbor is 1:

$$\mathbf{x}_{ij}^{norm} = \left[\frac{d_1(\mathbf{x}_{ij})}{r_i}, \ldots, \frac{d_p(\mathbf{x}_{ij})}{r_i}\right]^T \tag{4.16}$$

For simplification purpose, we denote in the following $\mathbf{x}_{ij}$ as $\mathbf{x}_{ij}^{norm}$. Fig. 4.4 illustrates the effect of neighborhood scaling in the pairwise space.

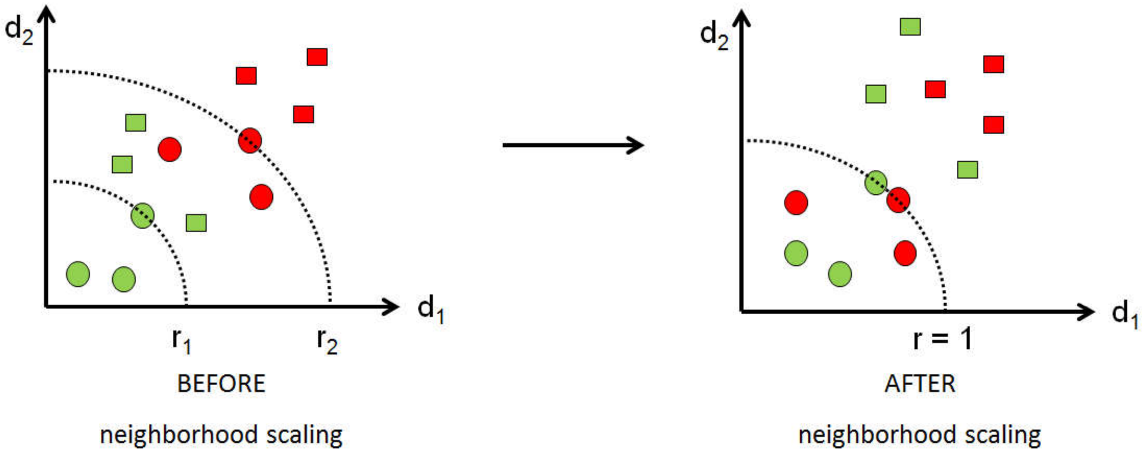

Figure 4.4: Effect of neighborhood scaling before (left) and after (right) on the neighborhood of two time series $\mathbf{x}_1$ (green) and $\mathbf{x}_2$ (red). Circle represent negative pairs (m-NN$^-$) and square represents positive pairs (m-NN$^+$) for $m = 2$ neighbors. Before scaling, the problem is not linearly separable. The spread of each neighborhood are not comparable. After scaling, the target neighborhood becomes comparable and in this example, the problem becomes linearly separable between the circles and the squares.

4.4 Definition of the dissimilarity measure

4.4.1 Support Vector Machine (SVM) resolution

Let $\{\mathbf{x}_{ij}; y_{ij} = \pm 1\}$, $\mathbf{x}_{ij} \in m$-NN$^+ \cup$ m-NN$^-$ be the training set, with $y_{ij} = +1$ for $\mathbf{x}_{ij} \in m$-NN$^+$ (same label) and -1 for $\mathbf{x}_{ij} \in m$-NN$^-$ (different labels). For a maximum margin between positive and negative pairs, the problem is formalized in an SVM framework as follows in the pairwise space $\mathscr{E}$:

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i,j}\xi_{ij} \tag{4.17}$$

$$\textbf{s.t.} \;\; y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \tag{4.18}$$

$$\xi_{ij} \geq 0 \tag{4.19}$$

Thanks to the unit radii normalization $\mathbf{x}_{ij}/r_i$, the SVM ensures a global large margin solution involving equally local neighborhood constraints (i.e. local margins).
In the linear case, a L_1 regularization in Eq. 4.17 leads to a sparse and interpretable $\mathbf{w}$ that uncovers the modalities, periods and scales that differentiate best positive from negative pairs for a robust nearest neighbors classification:

$$\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} ||\mathbf{w}||_1 + C\sum_{i,j}\xi_{ij} \tag{4.20}$$

The proposed M^{2}TML approach differs from the one of Time series Metric Learning (TML) by Linear/Quadratic programming (LP/QP) in which a SVM pairwise is used to learn the best weight vector $\mathbf{w}$ such that positive pairs are widely separated from negative pairs. Defining the learned metric D from the vector $\mathbf{w}$ needs to be careful.

4.4.2 Linear solutions

Let $\mathbf{x}_{test}$ be a new sample, $\mathbf{x}_{i,test} \in \mathscr{E}$ gives the proximity between $\mathbf{x}_i$ and $\mathbf{x}_{test}$ based on the p multi-modal and multi-scale metrics d_h. We denote $\mathbf{P_w}(\mathbf{x}_{i,test})$ the orthogonal projection of $\mathbf{x}_{i,test}$ on the axis of direction $\mathbf{w}$ and $||\mathbf{P_w}(\mathbf{x}_{i,test})||$ its norm that allows to measure the closeness between $\mathbf{x}_{test}$ and $\mathbf{x}_i$ while considering the discriminative features between positive and negative pairs. We review in this section different propositions to define the learned metric D: Scalar product, Projection Norm, Exponential transformation.

Problem linked to the SVM resolution

First, the learned metric D can be defined as the decision function obtained by solving the SVM problem:

$$D(\mathbf{x}_{i,test}) = \mathbf{w}^T\mathbf{x}_{i,test} + b \tag{4.21}$$

The obtained metric D doesn't necessarily satisfy the distinguishability ($D(\mathbf{x}_{ii} = 0)$) and positivity ($D(\mathbf{x}_{ij} \geq 0)$) property, especially when positive pairs (different classes) are situated nearer to the origin point than negative pairs (same class) (Fig. 4.5).

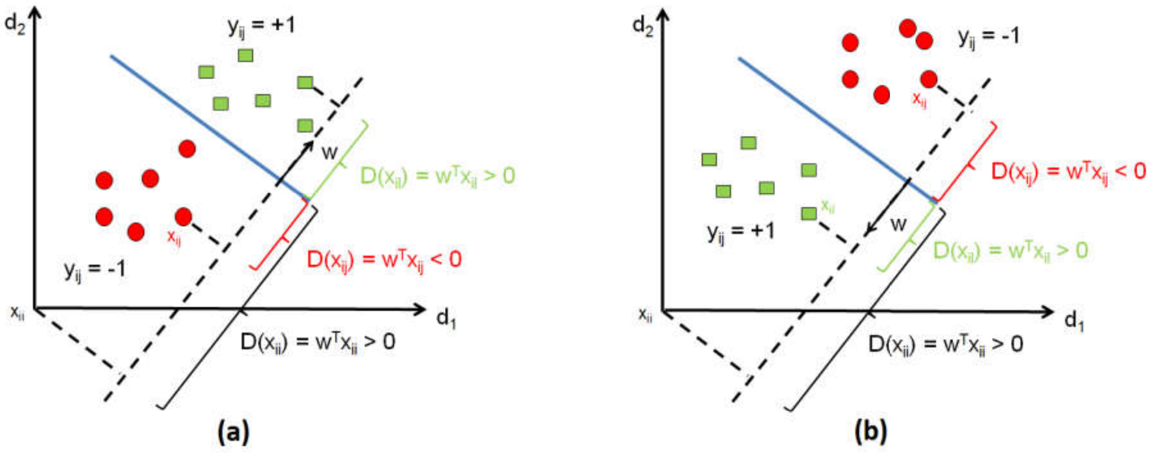
d$_2$
y$_{ij}$ = +1
w
D(x$_{ij}$) = w^{T}x$_{ij}$ > 0
x$_{ij}$
D(x$_{ij}$) = w^{T}x$_{ij}$ < 0
y$_{ij}$ = -1
x$_{ii}$
d$_1$
D(x$_{ii}$) = w^{T}x$_{ii}$ > 0
(a)
d$_2$
y$_{ij}$ = -1
x$_{ij}$
D(x$_{ij}$) = w^{T}x$_{ij}$ < 0
x$_{ij}$
y$_{ij}$ = +1
w
D(x$_{ij}$) = w^{T}x$_{ij}$ > 0
x$_{ii}$
d$_1$
D(x$_{ii}$) = w^{T}x$_{ii}$ > 0
(b)

Figure 4.5: Example of SVM solutions and of the resulting metric D defined by a scalar product. Fig. (a) represents common expected configuration where negative pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b), the vector $\mathbf{w} = [-1 - 1]$ indicates that positive pairs (y_{ij}) are on the side of the origin point. For the two configurations, two problems arises: First, for negative pairs, $D(\mathbf{x}_{ij}) \leq 0$. Secondly, for the origin point $\mathbf{x}_{ii}$, we obtain $D(\mathbf{x}_{ii}) \neq 0$.

The norm of the projection $\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})$ can be used to define the learned metric D as it measures the distance of the pair $\mathbf{x}_{i,test}$ from the origin point $\mathbf{x}_{ii}$ along to the direction $\mathbf{w}$:

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})|| = ||\mathbf{w}^T \mathbf{x}_{i,test}|| \tag{4.22}$$

Although the norm $||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})||$ satisfies metric properties, it doesn't always guarantee lower distances between negative pairs (same class) than positive pairs (different classes) as illustrated in Fig 4.6.

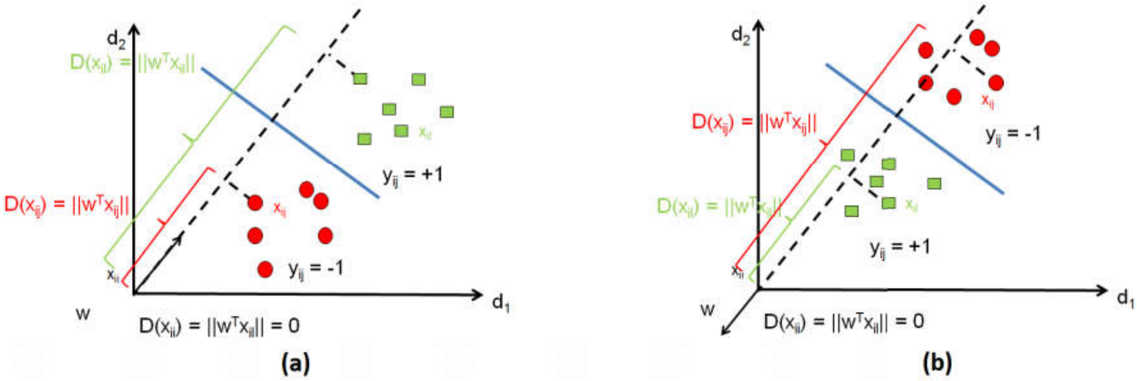

Figure 4.6: Example of SVM solutions and of the resulting metric D defined by the norm of the projection on $\mathbf{w}$. Fig. (a) represents common expected configuration where negative pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b), the vector $\mathbf{w} = [-1 \; -1]$ indicates that positive pairs (y_{ij}) are on the side of the origin point. One problem arrises in Fig. (b): distance of positive pairs $D(\mathbf{x}_{il})$ is lower than the distance of negative pairs $D(\mathbf{x}_{ij})$.

Exponential transformation

Secondly, we propose to add an exponential term to operate a "push" on negative pairs based on their distances to the separator hyperplan, that leads to the dissimilarity measure D of required properties:

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})||.\exp(\lambda[\mathbf{w}\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test}) + b]_{+}) \qquad \lambda > 0 \tag{4.23}$$

where λ controls the "push" term and $\mathbf{w}\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test}) + b$ defines the distance between the orthogonal projected vector and the separator hyperplane; $[t]_{+} = \max(0; t)$ being the positive operator. Note that, for a pair lying into the negative side ($y_{ij} = -1$), $[\mathbf{w}\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test}) + b]_{+} = 0$, the exponential term is vanished (i.e. no "pull" action) and the dissimilarity leads to the norm term. For a pair situated in the positive side ($y_{ij} = +1$), the norm is expanded by the push term, all the more the distance to the hyperplane is high.

Fig. 4.7, illustrates for $p = 2$ the behavior of the learned dissimilarity according to two extreme cases. The first one (Fig. 4.7-a), represents common expected configuration where negative pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin. The dissimilarity increases proportionally to the norm in the negative side, then exponentially on the positive side. Although the expansion operated in the positive side is dispensable in

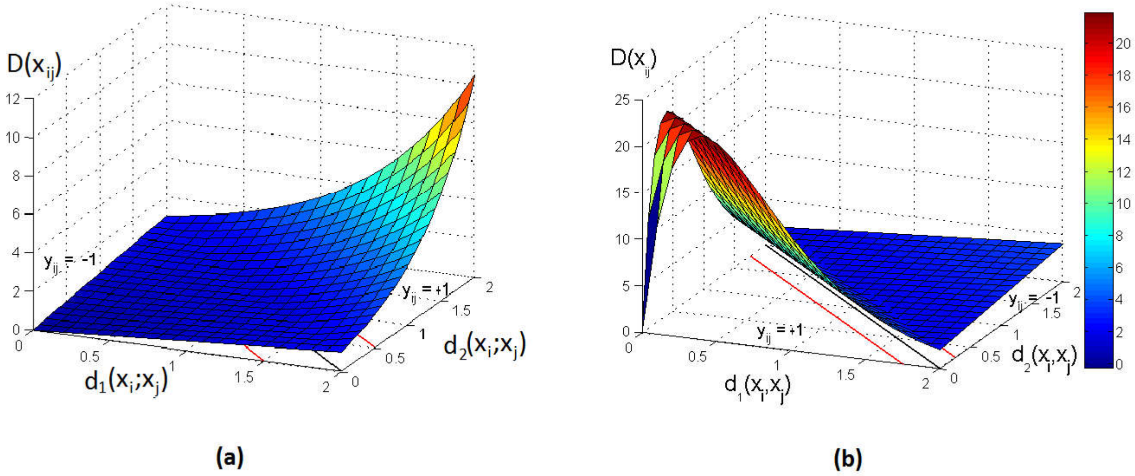

Figure 4.7: The behavior of the learned metric D ($p = 2$; $\lambda = 2.5$) with respect to common (a) and challenging (b) configurations of positive and negatives pairs.

that case, it doesn't affect nearest neighbors classification. Fig. 4.7-b, shows a challenging configuration where positive pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of different classes) are situated in the same side as the origin. That means that time series $\mathbf{x}_j$ that are of different classes from $\mathbf{x}_i$ are closer to $\mathbf{x}_i$ than its nearest neighbors. They are thus impostors. The dissimilarity behaves proportionally to the norm on the negative side, and increases exponentially from the hyperplane until an abrupt decrease induced by a norm near 0. Note that the region under the abrupt decrease mainly uncovers false positive pairs, i.e., pairs of norm zero labeled differently.

4.4.3 Non-linear solutions

The above solution holds true for any kernel K and allows to extend the dissimilarity D given in Eq. 4.23 to non linearly separable positive and negative pairs. Let K be a kernel defined in the pairwise space $\mathscr{E}$ and the related Hilbert space (feature space) $\mathscr{H}$. For a non linear combination function of the metrics $d_h, h = 1, \ldots, p$ in $\mathscr{E}$, we define the dissimilarity measure $D_{\mathscr{H}}$ in the feature space $\mathscr{H}$ as:

$$\begin{aligned} D_{\mathscr{H}}(\mathbf{x}_{i,test}) =& (||\mathbf{P}_{\mathbf{w}}(\phi(\mathbf{x}_{i,test}))|| - ||\mathbf{P}_{\mathbf{w}}(\phi(\mathbf{0}))||). \\ & \exp\left(\lambda \left[\sum_{ij} y_{ij}\alpha_{ij}K(\mathbf{x}_{ij}, \mathbf{x}_{i,test}) + b\right]_+\right) \qquad \lambda > 0 \end{aligned} \tag{4.24}$$

with $\phi(\mathbf{w})$ the image of $\mathbf{w}$ into the feature space $\mathscr{H}$ and the norm of the orthogonal projection of $\phi(\mathbf{x}_{i,test})$ on $\phi(\mathbf{w})$ as:

$$||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})|| = \frac{\sum_{ij} y_{ij}\alpha_{ij} K(\mathbf{x}_{ij}, \mathbf{x}_{i,test})}{\sqrt{\sum_{ijkl} \alpha_{ij}\alpha_{kl} y_{ij} y_{kl} K(\mathbf{x}_{ij}, \mathbf{x}_{kl})}} \tag{4.25}$$

Note that as $\phi(\mathbf{0})$ doesn't meet the origin in the feature space $\mathscr{H}$, the norms in Eq. 4.24 are centered with respect to $\phi(\mathbf{0})$.

We note that the proposed learned metric D and $D_{\mathscr{H}}$ are heuristic that solves the problem of positive pairs $\mathbf{x}_{il}$ on the side of the origin point $\mathbf{x}_{ii}$. Other solutions could have been proposed. In practice, the proposed D and $D_{\mathscr{H}}$ provides suitable solutions for our datasets.

4.5 Algorithms and extensions

Algorithm 1 summarizes the main steps to learn a multi-modal and multi-scale metric D for a robust nearest neighbors classification. Algorithm 2 details the steps to classify a new sample $\mathbf{x}_{test}$ using the learned metric D.

Note sur le neighborhood scaling en test?

Algorithm 1 Multi-modal and Multi-scale Temporal Metric Learning (M^{2}TML) for k-NN classification

```
Input: $X = \{\mathbf{x}_i, y_i\}_{i=1}^N$ $N$ labeled time series
           $d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 4.4
           a kernel $K$
Output: the learned dissimilarity $D$ or $D_{\mathscr{H}}$ depending of $K$
Pairwise embedding
    Embed pairs $(\mathbf{x}_i, \mathbf{x}_j)$ $i, j \in 1, ..., N$ into $\mathscr{E}$ as described in Eq. 4.5 and normalize $d_h$s
Build positive and negative pairs
    Build the sets of positive $m$-NN$^+$ and negative $m$-NN$^-$ pairs and scale the radii to 1 as described in 4.3
Train a SVM for a large margin classifier between $m$-NN$^+$ and $m$-NN$^-$ (Eq. 4.17)
Dissimilarity definition
    Consider Eq. 4.23 (resp. Eq. 4.24) to define $D$ (resp. $D_{\mathscr{H}}$) a linear (resp. non linear) combination function of the metrics $d_h$s.
```

Algorithm 1 can be easily extended for multivariate and regression problem. First, for multivariate problem, each unimodal metric d_h can be computed for each variable. Then, the above framework can be applied. For regression problem, the label y_i for each time series $\mathbf{x}_i$ is a continuous value. The only modification is at the neighborhood steps, when defining the positive and negative pairs labeled y_{ij}. For that, in Chapter 3, Section 3.1.1, we propose two different strategies to define the pairwise labels y_{ij}.

Algorithm 2 k-NN classification using the learned metric D or $D_{\mathscr{H}}$

```
Input: $X = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ $N$ labeled time series
       $\{\mathbf{x}_{test}, y_{test}\}$ a labeled time series to test
       $d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 4.4
       the learned dissimilarity $D$ or $D_{\mathscr{H}}$ depending of the kernel $K$
Output: Predicted label $\hat{y}_{test}$
Pairwise embedding
    Embed pairs $(\mathbf{x}_i, \mathbf{x}_{test})$ $i \in 1, ..., N$ into $\mathscr{E}$ as described in Eq. 4.5 and normalize $d_h$s using
    the same normalization parameters in Algorithm 1
Dissimilarity computation
    Consider Eq. 4.23 (resp. Eq. 4.24) to compute $D(\mathbf{x}_i, \mathbf{x}_{test})$ (resp. $D_{\mathscr{H}}(\mathbf{x}_i, \mathbf{x}_{test})$) a linear
    (resp. non linear) combination function of the metrics $d_h(\mathbf{x}_i, \mathbf{x}_{test})$.
Classification
    Consider the $k$ lowest dissimilarities $D(\mathbf{x}_i, \mathbf{x}_{test})$ (resp. $D_{\mathscr{H}}(\mathbf{x}_i, \mathbf{x}_{test})$). Extract the labels
    $y_i$ of the considered $\mathbf{x}_i$ and make a vote scheme to predict the label $\hat{y}_{test}$ of $\mathbf{x}_{test}$
```

4.6 Conclusion of the chapter

The adaptation of SVM in the pairwise space to learn a multi-modal and multi-scale metric D have brought us to propose a pre-processing step before solving the problem such as the neighborhood scaling, and a post-processing step such as defining the metric D as the objective of the SVM is to separate negative from positive classes.

Choosing a m-neighborhood, greater than the k-neighborhood, is used in the classifier SVM. It allows to limit the imposters to invade the neighborhood of the m-neighbors while controling the computation complexity.

As we have defined all functions components of our algorithms (learning, testing), we test our proposed algorithms M^{2}TML in the next part on standard datasets of the literature used for classification of univariate time series.