CHAPTER 3

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

Sommaire

In this chapter, we first motivate the problem of learning a multi-modal and multi-scale temporal metric for time series nearest neighbors classification. Secondly, we introduce the concept of dissimilarity space. Thirdly, we formalize the general problem of learning a combined metric. Then, we propose three different formalizations (Linear, Quadratic and SVM-based), each involving a different regularization term. We give an interpretation of the solution and study the properties of the obtained metric. Finally, we detail the retained solution and give the algorithm.

enlever les sous sections dans le sommaire

3.1 Motivations

The definition of a metric to compare samples is a fundamental issue in data analysis or machine learning. Contrary to static data, temporal data are more complex: they may be

compared not only on their amplitudes but also on their dynamic, frequential spectrum or other inherent characteristics. For time series comparison, a large number of metrics have been proposed, most of them are designed to capture similitudes and differences based on one temporal modality. For amplitude-based comparison, measures cover variants of Mahalanobis distance or the dynamic time warping (DTW) to cope with delays [BC94b]; [Rab89]; [SC78b]; [KL83]. Other propositions refer to temporal correlations or derivative dynamic time warping for behavior-based comparison [AT10b]; [RBK08]; [CCP06]; [KP01]; [DM09]. For frequential aspects, comparisons are mostly based on the Discret Fourier or Wavelet Transforms [SS12a]; [KST98]; [DV10]; [Zha+06]. A detailed review of the major metrics is proposed in [MV14]. In general, the most discriminant modality (amplitude, behavior, frequency, etc.) varies from a dataset to another.
Furthermore, in some applications, the most discriminative characteristic between time series of different classes can be localized on a smaller part of the signal. A crucial key to localize discriminative features is to define metrics that involves totally or partially time series elements rather than systematically the whole elements. In the most challenging applications, it appears that both factors (modality, scale) are needed to discriminate the classes. Some works propose to combine several modalities through a priori models as in [DCDG10]; [DCA12]; [SB08]. Fig. 3.1 shows an example of significant improvement in classification performances by taking into account in the metric definition, several modalities (amplitude d_A, behavior d_B, frequential d_F) located at different scales (illustrated in the figure). The performance of the learnt combined metric is compared with the ones of the standard metrics that take into account for each, only one modality on a global scale (involving all time series elements).

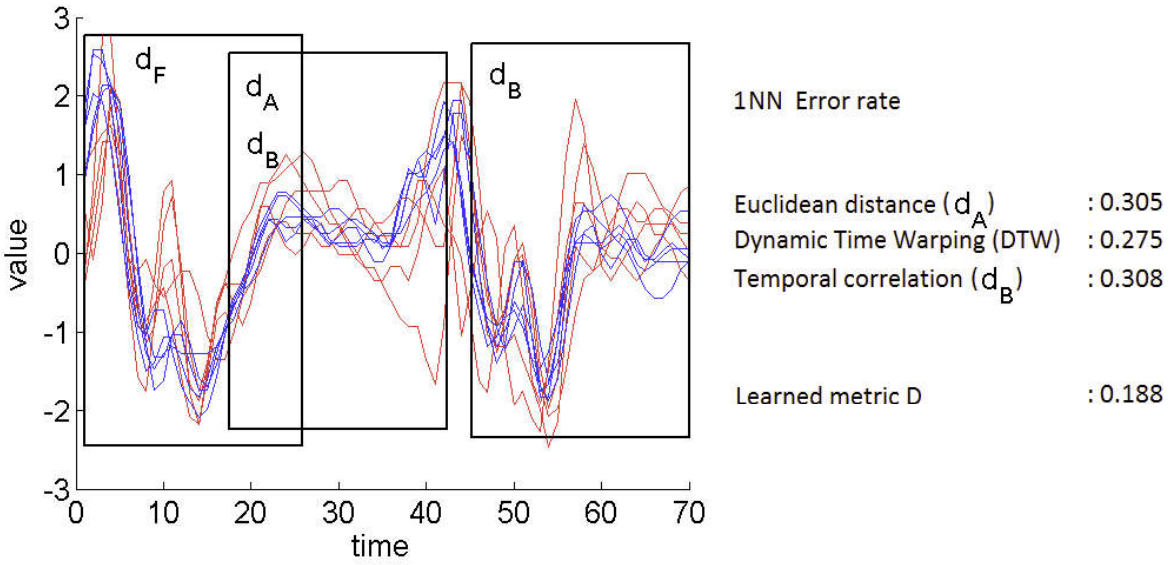

Figure 3.1: SonyAIBO dataset and error rate using a kNN ($k = 1$) with standard metrics (Euclidean distance, Dynamic Time Warping, temporal correlation) and a learned combined metric D. The figure shows the 4 major metrics involves in the combined metric D and their temporal scale (black rectangles).

Our aim is to take benefice of metric learning framework [WS09b]; [BHS12] to learn a

multi-modal and multi-scale temporal metric for time series nearest neighbors classification. Specifically, our objective is to learn from the data a linear or non linear function that combines several temporal modalities at several temporal scales, that satisfies metric properties (Section 2.2), and that generalizes the case of standard global metrics.
Metric learning can be defined as learning, from the data and for a task, a pairwise function (*i.e.* a similarity, dissimilarity or a distance) to make closer samples that are expected to be similar, and far away those expected to be dissimilar. Similar and dissimilar samples, are inherently task- and application-dependent, generally given *a priori* and fixed during the learning process. Metric learning has become an active area of research in last decades for various machine learning problems (supervised, semi-supervised, unsupervised, online learning) and has received many interests in its theoretical background (generalization guarantees) [BHS13]. From the surge of recent research in metric learning, one can identify mainly two categories: the linear and non linear approaches. The former is the most popular, it defines the majority of the propositions, and focuses mainly on the Mahalanobis distance learning [WS09a]. The latter addresses non linear metric learning which aims to capture non linear structure in the data. In Kernel Principal Component Analysis (KPCA) [ZY10]; [Cha+10], the aim is to project the data into a non linear feature space and learn the metric in that projected space. In Support Vector Metric Learning (SVML) approach [XWC12], the Mahalanobis distance is learned jointly with the learning of the SVM model in order to minimize the validation error. In general, the optimization problems are more expensive to solve, and the methods tends to favor overfitting as the constraints are generally easier to satisfy in a nonlinear kernel space. A more detailed review is done in [BHS13].
Contrary to static data, metric learning for structured data (*e.g.* sequence, time series, trees, graphs, strings) remains less numerous. While for sequence data most of the works focus on string edit distance to learn the edit cost matrix [OS06]; [BHS12], metric learning for time series is still in its infancy. Without being exhaustive, major recent proposals rely on weighted variants of dynamic time warping to learn alignments under phase or amplitude constraints [Rey11]; [JJO11]; [ZLL14], enlarging alignment learning framework to multiple temporal matching guided by both global and local discriminative features [FDCG13]. For the most of these propositions, temporal metric learning process is systematically: a) Unimodal (amplitude-based), the divergence between aligned elements being either the Euclidean or the Mahalanobis distance and b) Uni-scale (global level) by involving the whole time series elements, which restricts its potential to capture local characteristics. We believe that perpectives for metric learning, in the case of time series, should include multi-modal and multi-scale aspects.

We propose in this work to learn a multi-modal and multi-scale temporal metric for a robust k-NN classifier. For this, the main idea is to embed time series into a dissimilarity space [PPD02]; [DP12] where a linear function combining several modalities at different temporal scales can be learned, driven jointly by a SVM and nearest neighbors metric learning framework [WS09b]. Thanks to the "kernel trick", the proposed solution is extended to non-linear temporal metric learning context. A sparse and interpretable variant of the proposed metrics confirms its ability to localize finely discriminative modalities as well as their temporal scales. In the following, the term metric is used to reference both a distance or a dissimilarity measure.

In this chapter, we first present the concept of dissimilarity space. Then, we formalize the general problem of learning a combined metric for a robust k-NN as the learning a function in the dissimilarity space. From the general formalization, we propose three formalizations (Linear, Quadratic and SVM-based), give an interpretation of the solutions and study the properties of the learnt metrics. Finally, we detail the retained solution and give the algorithm. Note that these formalizations don't concern only time series and can be applied to any type of data.

3.2 Multi-modal and multi-scale dissimilarity representation

In this section, we first present the concept of dissimilarity space for multi-modal metrics. Then, in the case of time series, we enrich this representation with a multi-scale description.

Let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ be a set of n time series $\mathbf{x}_i = [x_{i1}, \ldots, x_{iq}] \in \mathbb{R}^q$ labeled y_i. Let $d_1, \ldots, d_p$ be p given metrics that allow to compare samples $\mathbf{x}_i$. As discussed in Chapter 2, three naturally modalities are involved for time series comparison: amplitude-based d_A, behavior-based d_B and frequential-based d_F. Our objective is to learn a metric D that combines the p basic temporal metrics for a robust k-NN classifier.

The computation of a metric d, and D, always takes into account a pair of samples $(\mathbf{x}_i, \mathbf{x}_j)$. We introduce a new space representation referred as the **dissimilarity space**. We note φ 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 $d_1, \ldots, d_p$ (Fig. 3.2):

$$\begin{aligned} \varphi : \mathbb{R}^q \times \mathbb{R}^q &\rightarrow \mathscr{E} \\ (\mathbf{x}_i, \mathbf{x}_j) &\rightarrow \mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), \ldots, d_p(\mathbf{x}_i, \mathbf{x}_j)]^T \end{aligned} \tag{3.1}$$

A metric D that combines the p metrics $d_1, \ldots, d_p$ can be seen as a function of the dissimilarity space: $D(\mathbf{x}_{ij}) = f(d_1(\mathbf{x}_i, \mathbf{x}_j), \ldots, d_p(\mathbf{x}_i, \mathbf{x}_j))$. In that space, the norm of a pairwise vector $||\mathbf{x}_{ij}||$ refers to the proximity between the time series $\mathbf{x}_i$ and $\mathbf{x}_j$. In particular, if $||\mathbf{x}_{ij}|| = 0$ then $\mathbf{x}_j$ is identical to $\mathbf{x}_i$ according to all metrics d_h. Note that in this space, the proximity between pairs can't be interpreted.

As illustrated in Fig. 3.1, the multi-modal representation in the dissimilarity space can be enriched for time series by measuring each unimodal metric d_h at different scales. Note that the distance measures (amplitude-based d_A, 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.

ref

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

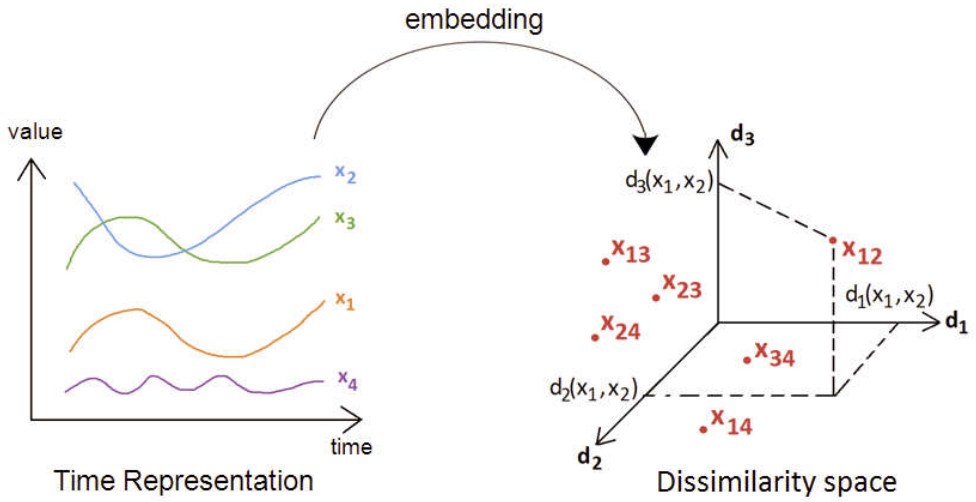

Figure 3.2: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the dissimilarity space (right) for $p = 3$ basic metrics.

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, covering discriminating subsequences in the central region of I (around $\frac{b-a}{2}$): $I = [a; b]; I_L = [a; a + \alpha(b - a)]; I_R = [a - \alpha(b - a); b]$. 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. 3.3. 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:

$$d_A^{Is}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t \in Is} (x_{it} - x_{jt})^2} \tag{3.2}$$

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

In the following, for simplification purpose, we consider $d_1, \ldots, d_p$ as the set of multi-modal and multi-scale metrics.

3.3 M²TML general problem

In this section, we propose to define the Multimodal and Multiscale Time series Metric Learning (M²TML) problem in the initial space as a general problem of learning a function in the dissimilarity space. First, we give the intuition and formalize the general optimization problem. Secondly, we propose different strategies to define the neighborhood.

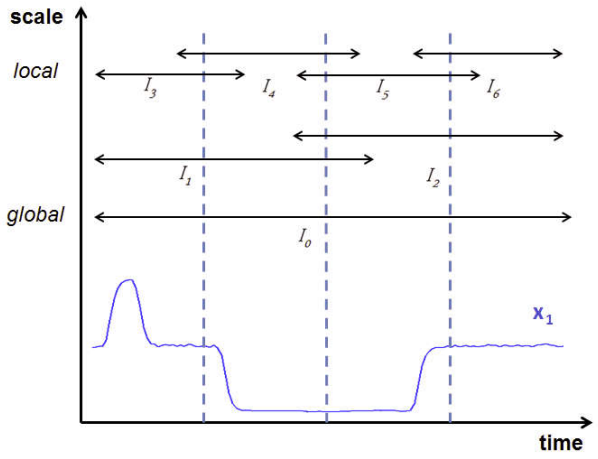

Figure 3.3: Multi-scale decomposition

3.3.1 General formalization for M^{2}TML

Our objective is to learn a dissimilarity $D = f(d_1, \ldots, d_p)$ in $\mathscr{E}$, the embedding space, that combines the p dissimilarities $d_1, \ldots, d_p$ for a robust k-NN classifier. The function f can be linear or non-linear and must satisfy at least the properties of a dissimilarity, *i.e.*, positivity, reflexivity and symmetry [DD09]. To simplify the discussion in the following, we refer to dissimilarity as metrics. The proposition is based on two standard intuitions in metric learning, *i.e.*, for each time series $\mathbf{x}_i$, the metric D should bring closer the time series $\mathbf{x}_j$ of the same class ($y_j = y_i$) while pushing the time series $\mathbf{x}_l$ of different classes ($y_l \neq y_i$). These two sets are called respectively $Pull_i$ and $Push_i$. Fig. 3.4 illustrates that concept. Formally, the metric learning problem can be written as an optimization problem that involves both a regularized $R(Pull)$ and a loss term $L(Push)$ under constraints that controls the push term:

$$\begin{aligned} &\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left[R(Pull) + L(Push)\right] \\ &\text{s.t. } \textit{constaints} \end{aligned} \tag{3.3}$$

The problem of learning a combined metric D can be written as the following optimization problem:

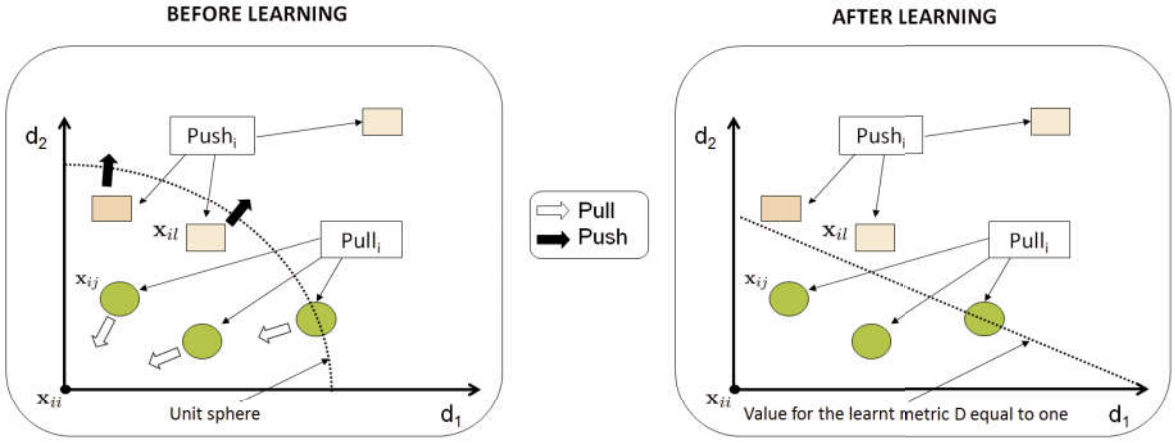

Figure 3.4: Geometric representation of the metric learning problem in the dissimilarity space for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), push samples $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the dissimilarity space, this is equivalent to have pairwise vectors $\mathbf{x}_{il} \in Push_i$ with a norm lower to some pairwise target $\mathbf{x}_{ij} \in Pull_i$. The aim of metric learning is to push pairwise $\mathbf{x}_{il}$ (black arrow) and pull pairwise $\mathbf{x}_{ij}$ from the origin (white arrow).

$$\begin{aligned} \underset{D,\xi}{\operatorname{argmin}} \Bigg\{ & \underbrace{\sum_i \sum_{j \in Pull_i} D(\mathbf{x}_{ij})}_{pull} + C \underbrace{\sum_i \sum_{\substack{j \in Pull_i \\ l \in Push_i}} \frac{1 - y_{il}}{2} \xi_{ijl}}_{push} \Bigg\} \\ \text{s.t. } & \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ & D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \end{aligned} \tag{3.4}$$

where $y_{il} = -1$ if $y_i \neq y_l$ and $+1$ otherwise, ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs. In the next section, we detailed different strategies to define the $Pull_i$ and $Push_i$ sets.

3.3.2 Push and pull set definition

To build the pairwise training set, we propose at least three solutions, illustrated in Fig 3.5, but other propositions are possible:

1. **k-NN vs impostors**: it corresponds to the union for all $\mathbf{x}_i$, of the set of all pairs $\mathbf{x}_{ij}$ ($y_j = y_i$) such $||\mathbf{x}_{ij}||$ are the k-lowest norms (target set) and of all pairs $\mathbf{x}_{il}$ ($y_l \neq y_i$) that invades the neighborhood defined the pairs $\mathbf{x}_{ij}$:

$$\forall i \in 1,\ldots,n, \quad Pull_i = \{\mathbf{x}_{ij} \text{ s.t. } y_j = y_i \text{ and } ||\mathbf{x}_{ij}||_2 \text{ are the } k\text{-lowest norms}\} \tag{3.5}$$
$$Push_i = \{\mathbf{x}_{il} \text{ s.t. } y_l \neq y_i, ||\mathbf{x}_{il}||_2 \leq \max_{\mathbf{x}_{ij} \in Pull_i} ||\mathbf{x}_{ij}||_2\} \tag{3.6}$$

2. **k-NN vs all**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and of all pairs $\mathbf{x}_{il}$ such that $y_l \neq y_i$. It ensures that no pairs $\mathbf{x}_{il}$ will invade the target neighborhood during the learning process:

$$\forall i \in 1,\ldots,n, \quad Pull_i = \{\mathbf{x}_{ij} \text{ s.t. } y_j = y_i \text{ and } ||\mathbf{x}_{ij}||_2 \text{ are the } k\text{-lowest norms}\} \tag{3.7}$$
$$Push_i = \{\mathbf{x}_{il} \text{ s.t. } y_l \neq y_i\} \tag{3.8}$$

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 ($y_j = y_i$), 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$_i^-$. More precisely, our proposition states: $m = \alpha.k$ with $\alpha \geq 1$. Other propositions for m are possible:

$$\forall i \in 1,\ldots,n, \quad Pull_i = \{\mathbf{x}_{ij} \text{ s.t. } y_j = y_i \text{ and } ||\mathbf{x}_{ij}||_2 \text{ are the } m\text{-lowest norms}\} \tag{3.9}$$
$$Push_i = \{\mathbf{x}_{il} \text{ s.t. } y_l \neq y_i \text{ and } ||\mathbf{x}_{il}||_2 \text{ are the } m\text{-lowest norm}\} \tag{3.10}$$

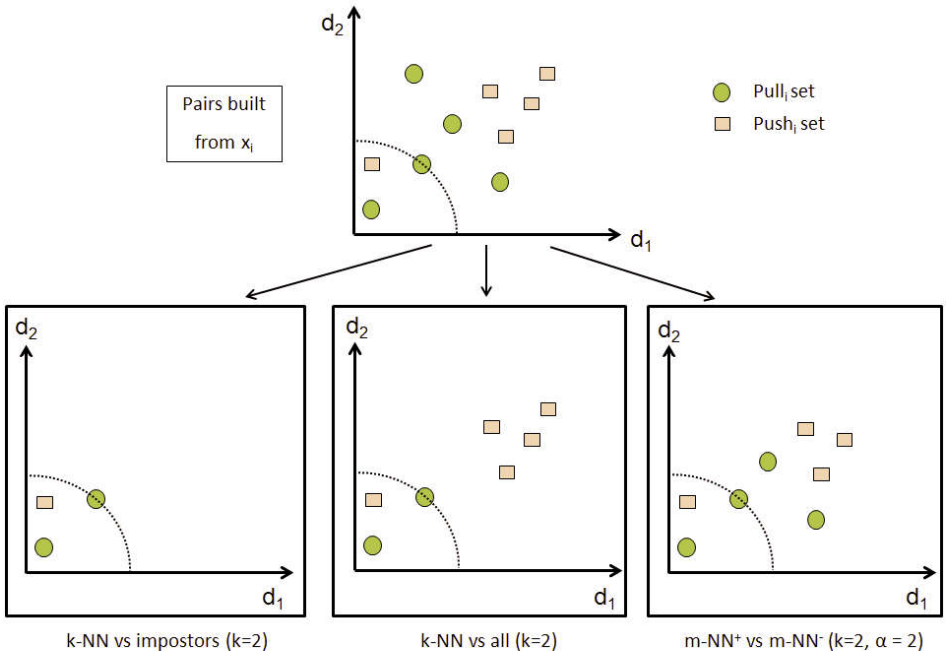

Figure 3.5: Example of different strategies to build $Pull_i$ and $Push_i$ sets for a $k = 2$ neighborhood.

Finally, let discuss about the similarities and differences between LMNN (Weinberger & Saul [WS09a]) and our M^{2}TML proposition. In LMNN, the sets $Pull_i$ and $Push_i$ are defined according the **k-NN vs impostors** strategy (Eqs. 3.5 & 3.6) and may be unbalanced. The

sets are defined and fixed during the optimization process according to an initial metric (Euclidean distance). In M²TML the sets $Pull_i$ and $Push_i$ are defined according the k-**NN vs impostors** strategy (Eqs. 3.9 & 3.10) and are balanced. The sets are defined and fixed during the optimization process according to an initial metric (Euclidean distance), but the m-neighborhood is larger than the k-neighborhood. By considering a neighborhood larger than the k-neighborhood (k-**NN vs impostors** strategy), we believe that the generalization properties of the learnt metric D would be improved.

In the following, we propose different regularizers for the pull term $R(Pull)$. First, we use a linear regularization. Secondly, we use a quadratic regularization that enables to extend the method to learn non-linear function for D by using the "kernel" trick. Thirdly, we formulate the problem as a SVM problem to solve a large margin problem between $Pull_i$ and $Push_i$ sets, and then, induce a combined metric D. Finally, we sum up the retained solution (SVM-based solution) and give the main steps of the algorithm.

3.4 Linear formalization for M²TML

In this section, we define the problem of learning a combined metric D as a linear combination in the dissimilarity space using a linear regularizer. First, we give the optimization problem. Then, we discuss on the properties of the learnt metric D.
Let $\mathbf{X} = \{\mathbf{x}_{ij}, y_{ij}\}_{i,j=1}^{n}$ be a set of pairwise vectors $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$ described by p metrics $d_1, \ldots, d_p$ and labeled $y_{ij} = -1$ if $y_i \neq y_j$ and $+1$ otherwise. We consider a linear combination of the p metrics:

$$D(\mathbf{x}_{ij}) = \mathbf{w}^T \mathbf{x}_{ij} = \sum_{h=1}^{p} w_h . d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{3.11}$$

where $\mathbf{w} = [w_1, \ldots, w_p]^T$ is the vector of weights w_h. From Eq. 3.4, learning a linear combined metric D can be formalized as follow:

$$\begin{aligned} \underset{\mathbf{w},\xi}{\operatorname{argmin}} & \left\{ \underbrace{\sum_{\substack{i \\ j \in Pull_i}} \mathbf{w}^T \mathbf{x}_{ij}}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1 - y_{il}}{2} \xi_{ijl}}_{push} \right\} \\ \text{s.t. } & \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ & \mathbf{w}^T (\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \\ & \forall h = 1, \ldots, p, \quad w_h \geq 0 \end{aligned} \tag{3.12}$$

where ξ_{ijl} are the slack variables, C the trade-off between the pull and push costs, and $Pull_i$ and $Push_i$ are defined in Eqs. 3.9 & 3.10. Similarly to SVM, note that the slack variables ξ_{ijl} can be interpreted. In particular, the pairs $\mathbf{x}_{ij}$ and $\mathbf{x}_{il}$ that violate the constraints ($\mathbf{w}^T\mathbf{x}_{il} < \mathbf{w}^T\mathbf{x}_{ij}$) will be penalized in the objective function. They corresponds to push pairs $\mathbf{x}_{il}$ that invades the neighborhood of the pull pairs $\mathbf{x}_{ij}$.

The problem is very similar to a C-SVM classification problem. When C is infinite, we have a "strict" problem: the solver will try to find a direction $\mathbf{w}$ in the dissimilarity space $\mathscr{E}$ for which all $\xi_{ijl} = 0$, that means that only pull samples should be in the close neighborhood of each $\mathbf{x}_i$. Let denote $\mathbf{x}^*_{ij}$ and $\mathbf{x}^*_{il}$, the vectors for which $\xi_{ijl} = 0$. In that case, if a solution is found, the margin $\min_{i,j,l}(||\mathbf{x}^*_{il} - \mathbf{x}^*_{ij}||_2)$ can be derived from the tightest constraint, for which equality holds:

$$\begin{aligned}
\mathbf{w}^T(\mathbf{x}^*_{il} - \mathbf{x}^*_{ij}) &= 1 \\
\mathbf{w}^T(\mathbf{x}^*_{il} - \mathbf{x}^*_{ij})^T(\mathbf{x}^*_{il} - \mathbf{x}^*_{ij})\mathbf{w} &= 1 \\
||\mathbf{x}^*_{il} - \mathbf{x}^*_{ij}||_2^2 &= \frac{1}{||\mathbf{w}||_2^2} \\
||\mathbf{x}^*_{il} - \mathbf{x}^*_{ij}||_2 &= \frac{1}{||\mathbf{w}||_2}
\end{aligned}$$

Concerning the properties of D, positivity is ensured with the constraints $w_h \geq 0$ and because $d_1, \ldots, d_p$ are dissimilarity measures ($d_h \geq 0$). As the metric D is defined as a linear combination of dissimilarity measures $d_1, \ldots, d_p$, it can be shown that symmetry and reflexivity is verified.

3.5 Quadratic formalization for M^{2}TML

In this section, we define the problem of learning D as a linear or non-linear combination in the dissimilarity space using a quadratic regularizer. First, we give the optimization problem and its dual formulation form involving only dot products. Then, we discuss on the properties of the learnt metric D. Finally, we study a link between SVM and the quadratic formalization.

3.5.1 Primal and dual formalization

The formulation in Eq. 3.12 suppose that the metric D is a linear combination of the metrics d_h. The linear formalization being similar to the one of SVM, it can be derived into its dual form to extend the method to find non-linear solutions for D. For that, we propose to change

the linear regularizer $R(Pull)$ in the objective function of Eq. 3.12 into a quadratic regularizer:

$$\begin{aligned} &\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ R(Pull) + C \sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1 - y_{il}}{2} \xi_{ijl} \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &\quad \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ &\quad \xi_{ijl} \geq 0 \end{aligned} \tag{3.13}$$

Two solutions for $R(Pull)$ are studied:

1. $$R(Pull) = \frac{1}{2} \sum_{h=1}^{p} \sum_{\substack{i \\ j \in Pull_i}} (w_h d_h(\mathbf{x}_{ij}))^2 \tag{3.14}$$

2. $$R(Pull) = \frac{1}{2} \sum_{h=1}^{p} \left(\sum_{\substack{i \\ j \in Pull_i}} w_h d_h(\mathbf{x}_{ij}) \right)^2 = \frac{1}{2} m.n \sum_{h=1}^{p} \left(w_h \bar{d}_h \right)^2 \tag{3.15}$$

where $\bar{d}_h = \frac{1}{mn} \sum_{\substack{i \\ j \in Pull_i}} d_h(\mathbf{x}_{ij})$ denotes the mean of the distances $d_h(\mathbf{x}_{ij})$ for each metric d_h. Let $\bar{\mathbf{x}} = [\bar{d}_1, \ldots, \bar{d}_p]^T$ be a vector of size p containing the mean of the metrics $\bar{d}_1, \ldots, \bar{d}_p$.

Other regularizations are possible. We focus on these two propositions that can be reduced to the following formula:

$$R(Pull) = \frac{1}{2} \mathbf{w}^T \mathbf{M} \mathbf{w} \tag{3.16}$$

where $\mathbf{M}$ denotes respectively the following matrix for each regularizer:

1. $$\mathbf{M} = Diag(\mathbf{X}_{pull}^T \mathbf{X}_{pull}) = \begin{bmatrix} \sum\limits_{\substack{i \\ j \in Pull_i}} d_1^2(\mathbf{x}_{ij}) & & 0 \\ & \ddots & \\ 0 & & \sum\limits_{\substack{i \\ j \in Pull_i}} d_p^2(\mathbf{x}_{ij}) \end{bmatrix} \tag{3.17}$$

2. $$\mathbf{M} = Diag(\bar{\mathbf{x}}) Diag(\bar{\mathbf{x}}) = \begin{bmatrix} \bar{d}_1{}^2 & & 0 \\ & \ddots & \\ 0 & & \bar{d}_p{}^2 \end{bmatrix} \tag{3.18}$$

where $\mathbf{X}_{pull} = \bigcup_i Pull_i$ be a $(m.n) \times p$ matrix containing the vector $\mathbf{x}_{ij} \in Pull_i$.

From this, the optimization problem can be written using a quadratic regularization for the pull term:

$$\begin{aligned}
&\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w}}_{pull} + C \underbrace{\sum_{\substack{i \\ j\in Pull_i \\ l\in Push_i}} \frac{1-y_{il}}{2}\xi_{ijl}}_{push} \right\} \\
&\text{s.t. } \forall i = 1,\dots,n, \forall j \in Pull_i, l \in Push_i, \\
&\quad \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\
&\quad \xi_{ijl} \geq 0
\end{aligned} \tag{3.19}$$

Similarly to SVM, this formulation can be reduced to the minimization of the following Lagrange function $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$, consisting of the sum of the objective function and the constraints multiplied by their respective Lagrange multipliers $\boldsymbol{\alpha}$ and $\mathbf{r}$:

$$\begin{aligned}
L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) =& \frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w} + C\sum_{ijl}\frac{1-y_{il}}{2}\xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\
&- \sum_{ijl}\alpha_{ijl}\left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl})\right)
\end{aligned} \tag{3.20}$$

where $\alpha_{ijl} \geq 0$ and $r_{ijl} \geq 0$ are the Lagrange multipliers. At the minimum value of $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$, we assume the derivatives with respect to $\mathbf{w}$ and ξ_{ijl} are set to zero:

$$\begin{aligned}
\frac{\partial L}{\partial \mathbf{w}} &= \mathbf{M}\mathbf{w} - \sum_{ijl}\alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) = 0 \\
\frac{\partial L}{\partial \xi_{ijl}} &= C - \alpha_{ijl} - r_{ijl} = 0
\end{aligned}$$

The matrix $\mathbf{M}$ being diagonal in both case (Eqs. 3.17 & 3.18), it is thus inversible. The equations leads to:

$$\mathbf{w} = \mathbf{M}^{-1}\sum_{ijl}\alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.21}$$

$$r_{ijl} = C - \alpha_{ijl} \tag{3.22}$$

Substituting Eq. 3.21 and 3.22 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.20, we get the dual formulation[1]:

[1] complete details of the calculations in Appendix D

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1} (\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &0 \leq \alpha_{ijl} \leq C \end{aligned} \tag{3.23}$$

For any new pair of samples $\mathbf{x}_{i'}$ and $\mathbf{x}_{j'}$, the resulting metric D writes:

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1}}_{\mathbf{w}^T} \mathbf{x}_{i'j'} \tag{3.24}$$

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{il}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Push \text{ set}} - \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{ij}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Pull \text{ set}} \tag{3.25}$$

3.5.2 Non-linear combined metric

The above formula can extended to non-linear function for the metric D. The inner product in Eq. 3.25 can be easily kernelized using the "kernel" trick:

$$\begin{aligned} \mathbf{x}_{il}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'} &= \mathbf{x}_{il}^T \mathbf{M}^{-\frac{1}{2}} \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'} \\ &= \left(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}\right)^T \left(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'}\right) \\ &= < \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'} > \\ &= \kappa(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'}) \end{aligned}$$

As $\mathbf{M}^{-1}$ is a diagonal matrix, it is inversible and can be written $\mathbf{M}^{-1} = \mathbf{M}^{-\frac{1}{2}} \mathbf{M}^{-\frac{1}{2}}$. For each regularization, we give below the matrix $\mathbf{M}^{-\frac{1}{2}}$:

1. $$\mathbf{M}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sum_{j \in Pull_i}^{i} d_1^2(\mathbf{x}_{ij})} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sum_{j \in Pull_i}^{i} d_p^2(\mathbf{x}_{ij})} \end{bmatrix} \tag{3.26}$$

2. $$\mathbf{M}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\bar{d}_1^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\bar{d}_p^2} \end{bmatrix} \tag{3.27}$$

The matrix $\mathbf{M}^{-\frac{1}{2}}$ can be interpreted in the first regularization proposition as a normalization

by the variance of the distance for each metric d_h. In the second regularization, it can be interpreted as a normalization by the mean of the distance for each metric d_h.

By replacing the inner product by a kernel back into Eq. 3.25, we obtain:

$$D(\mathbf{x}_{i'j'}) = \overbrace{\sum_{ijl} \alpha_{ijl} \kappa(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Push \text{ set}} - \overbrace{\sum_{ijl} \alpha_{ijl} \kappa(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Pull \text{ set}} \tag{3.28}$$

Let's give some interpretation and discussion about the properties of D. Similarly to SVM, from Eq. 3.24, at the optimality, only the triplets $(\mathbf{x}_{il} - \mathbf{x}_{ij})$ with $\alpha_{ijl} > 0$ are considered as the support vectors and the computation of the metric D depends only on these support vectors. Note that in this case, there exists two categories of support vectors (Eqs. 3.25 & 3.28): the vectors $\mathbf{x}_{il}$ from the push set $Push_i$ and the vectors $\mathbf{x}_{ij}$ from the pull set $Pull_i$ which $\alpha_{ijl} > 0$. The resulting metric D can be interpreted as the difference involving two similarity terms: a new pair $\mathbf{x}_{i'j'}$ is as dissimilar as its similarity to the $Push$ set is high while its similarity to the $Pull$ set is low. Inversely, the pair $\mathbf{x}_{i'j'}$ is as similar as its similarity to the $Push$ set is low while its similarity to the $Pull$ set is high.

Concerning the properties of the metric D, it can be shown that symmetry and reflexivity is ensured. However, as D is a difference of two similarity terms, positivity is not always ensured, *e.g.*, the similarity of the pull term is greater than the similarity of the push term. The resulting metric D is not thus a dissimilarity measure.

3.5.3 Link between SVM and the quadratic formalization

Many parallels have been studied between Large Margin Nearest Neighbors (LMNN) and SVM (Section 2.6.3). SVM is a well known framework: its has been well implemented in many libraries (*e.g.*, LIBLINEAR [FCH08] and LIBSVM [HCL08]), well studied for its generalization properties and extension to non-linear solutions. Similarly, we can make a link between the quadratic formalization and a SVM problem where the form of the metric D is defined *a priori*.

We show in Appendix E that solving the SVM problem in Eq. 3.29 for $\mathbf{w}$ and b solves a similar problem with a quadratic regularization in Eq. 3.19 for $D(\mathbf{x}_{ij}) = -\frac{1}{2}(\mathbf{w}^T\mathbf{x}_{ij} + b)$.

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i \text{ or} \\ j \in Push_i}} p_i \xi_{ij} \right\} \\ &\text{s.t. } \forall i, j \in Pull_i \text{ or } j \in Push_i : \\ &y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \end{aligned} \tag{3.29}$$

where p_i is a weight factor for each slack variable ξ_{ij} (in classical SVM, $p_i = 1$). The loss part

in the SVM formulation can be split into 2 terms involving the sets $Pull_i$ and $Push_i$:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i}} p_i^+ \xi_{ij} + C \sum_{\substack{i \\ l \in Push_i}} p_i^- \xi_{il} \right\} \\ &\text{s.t.} \ : \\ &\forall i, j \in Pull_i : y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\forall i, l \in Push_i : y_{il}(\mathbf{w}^T \mathbf{x}_{il} + b) \geq 1 - \xi_{il} \end{aligned} \tag{3.30}$$

where p_i^+ and p_i^- are the weight factors for pull pairs $Pull_i$ and push pairs $Push_i$. To obtain an equivalence, we set p_i^- as the half of the number pairs in $Pull_i$ and p_i^+ as the half of the number of time series L in $Push_i$:

$$p_i^- = \frac{k}{2} = \sum_{j \in Pull_i} \frac{1}{2} \tag{3.31}$$

$$p_i^+ = \frac{L}{2} = \frac{1}{2} \sum_{l} \frac{1 - y_{il}}{2} \tag{3.32}$$

In particular, let's underline the main similarities and differences:

- Both problems share a same set of constraints between triplets:

$$\forall i, j \in Pull_i, l \in Push_i : D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl}$$

- The SVM problem includes an additional set of constraints that is not present in the quadratic formalization. SVM takes into account pull pairs $\mathbf{x}_{ij}$ and push pairs $\mathbf{x}_{kl}$ that doesn't belong to the same neighborhood:

$$\forall i, j \in Pull_i, k, l \in Push_k, i \neq k : D(\mathbf{x}_{kl}) - D(\mathbf{x}_{ij}) \geq 1 - \frac{\xi_{kl} + \xi_{ij}}{2}$$

- The SVM problem includes in the loss term additional slack variables ξ_{ijl} that are not present in the quadratic formalization because of the additional set of constraints. It is not only a push term.

- The two problems involves different regularized/pull term: in SVM, the regularizer $\frac{1}{2}||\mathbf{w}||_2^2$ only involves the weight vector $\mathbf{w}$, whereas in the quadratic formalization, the regularizer involves also the pull pairs (Eqs. 3.14 & 3.15)

- Both problem suppose at first a linear combination for D.

Concerning the properties of the metric D, positivity is not ensured as SVM tries to find an hyperplane, the constraint $w_h \geq 0$ does not hold. Symmetry and reflexivity is ensured.

3.6 SVM-based formalization for M^{2}TML

In this section, we present a solution based on SVM where the form of the metric D is not known *a priori*. We formulate the problem as a SVM problem to solve a large margin problem between $Pull_i$ and $Push_i$ sets, and then, induce a combined metric D for the obtained SVM solution. Thanks to the SVM framework, the proposition can be naturally extended to learn both, linear or non-linear function for the metric D.

3.6.1 Support Vector Machine (SVM) resolution

Let $\{\mathbf{x}_{ij}; y_{ij} = \pm 1\}$, $\mathbf{x}_{ij} \in Pull_i \cup Push_i$ be the training set, with $y_{ij} = -1$ for $\mathbf{x}_{ij} \in Push_i$ and $+1$ for $\mathbf{x}_{ij} \in Pull_i$. For a maximum margin between the sets $Pull_i$ and $Push_i$, the problem is formalized in the dissimilarity space $\mathscr{E}$:

$$\begin{aligned} &\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i,j}\xi_{ij} \\ &\textbf{s.t. } y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\xi_{ij} \geq 0 \end{aligned} \tag{3.33}$$

In the linear case, a L_1 regularization in Eq. 3.33 leads to a sparse and interpretable $\mathbf{w}$ that uncovers the modalities, periods and scales that differentiate best pull from push pairs for a robust nearest neighbors classification. Note that in practice, the local neighborhoods for each sample $\mathbf{x}_i$ can have very different scales. Thanks to the unit radii normalization $\mathbf{x}_{ij}/r_i$, where r_i denotes the norm of the m-th neighbors in $Pull_i$, the SVM ensures a global large margin solution involving equally local neighborhood constraints (*i.e.* local margins).

3.6.2 Linearly separable Pull and Push sets

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 $\mathbf{P}_\mathbf{w}(\mathbf{x}_{i,test})$ the orthogonal projection of $\mathbf{x}_{i,test}$ on the axis of direction $\mathbf{w}$ and $||\mathbf{P}_\mathbf{w}(\mathbf{x}_{i,test})||$ its norm that allows to measure the closeness between $\mathbf{x}_{test}$ and $\mathbf{x}_i$ while considering the discriminative features between pull and push pairs. We review in this section different interpretations in the dissimilarity space.

Distance to the margin and Projected norm
Given a test pair $\mathbf{x}_{i,test}$, the distance to the margin $\mathbf{w}^T\mathbf{x}_{i,test} + b$ is a signed quantity. It doesn't verify the positivity property and can't thus be used to define the metric D. Secondly, we recall the formula of the projection of a pair $\mathbf{x}_{i,test}$ on the vector $\mathbf{w}$:

$$\mathbf{P}_\mathbf{w}(\mathbf{x}_{i,test}) = \frac{< \mathbf{w}, \mathbf{x}_{i,test} >}{||\mathbf{w}||^2}\mathbf{w} = \frac{\mathbf{w}^T\mathbf{x}_{i,test}}{||\mathbf{w}||^2}\mathbf{w} \tag{3.34}$$

The norm of the projection, measures the distance of the pair $\mathbf{x}_{i,test}$ from the origin point $\mathbf{x}_{ii}$ and takes into account the discriminative features:

$$||\mathbf{P_w}(\mathbf{x}_{i,test})|| = ||\mathbf{w}^T\mathbf{x}_{i,test}||/||\mathbf{w}|| \tag{3.35}$$

As illustrated in Fig. 3.6, by considering the norm of the projection, a pair $\mathbf{x}_{ij}$ becomes as similar as a pair $\mathbf{x}_{ij'}$ whereas their norm $||\mathbf{x}_{ij}||$ and $||\mathbf{x}_{ij'}||$ shows their are not similar.

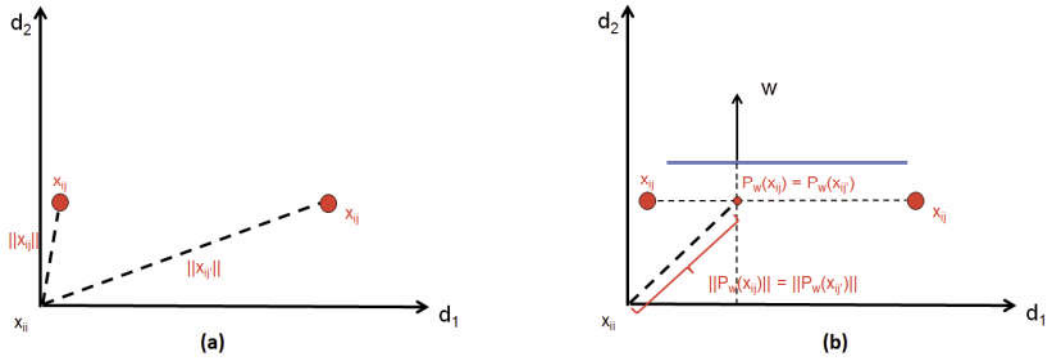

Figure 3.6: Difference between the norm of a vector (Fig. a) and the projected norm (Fig. b)

Although the norm $||\mathbf{P_w}(\mathbf{x}_{i,test})||$ satisfies the positivity metric properties, it doesn't always guarantee lower distances between pull pairs $Pull_i$ than push pairs $Push_i$ as illustrated in Fig 3.7.

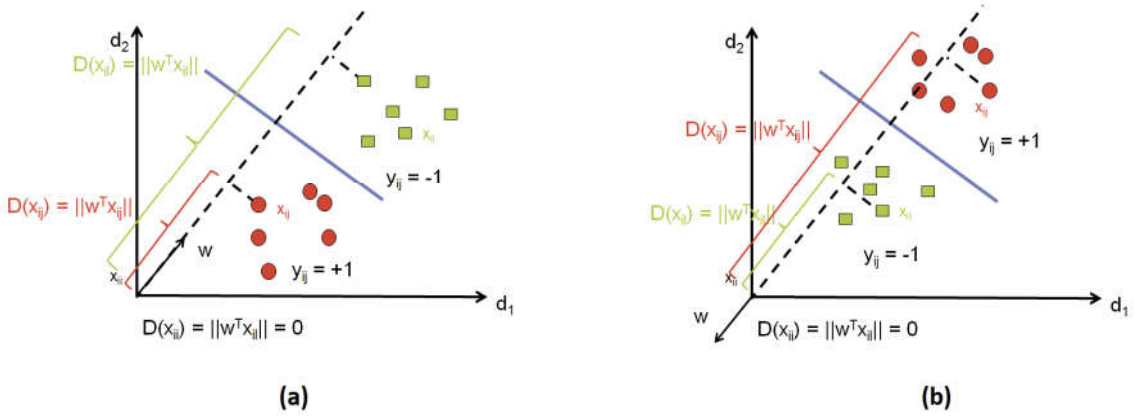

Figure 3.7: Example of SVM solutions and of the resulting metric D defined by the norm of the projection on $\mathbf{w}$. Fig. (a) represents common expected configuration where pull pairs $Pull_i$ are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b), the vector $\mathbf{w} = [-1 - 1]^T$ indicates that push pairs $Push_i$ are on the side of the origin point. One problem arises in Fig. (b): distance of push pairs $D(\mathbf{x}_{il})$ is lower than the distance of pull pairs $D(\mathbf{x}_{ij})$.

M²TML metric definition

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

properties:

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P_w}(\mathbf{x}_{i,test})||. \exp(\lambda[-(\mathbf{wP_w}(\mathbf{x}_{i,test}) + b)]_+) \quad \lambda > 0 \tag{3.36}$$

where λ controls the "push" term and $\mathbf{wP_w}(\mathbf{x}_{i,test}) + b$ 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 pull side ($y_{ij} = +1$), $[-(\mathbf{wP_w}(\mathbf{x}_{i,test}) + b)]_+ = 0$, the exponential term is vanished (i.e. no "pull" action) and the dissimilarity leads to the norm term. For a pair situated in the push side ($y_{ij} = -1$), the norm is expanded by the push term, all the more the distance to the hyperplane is high.

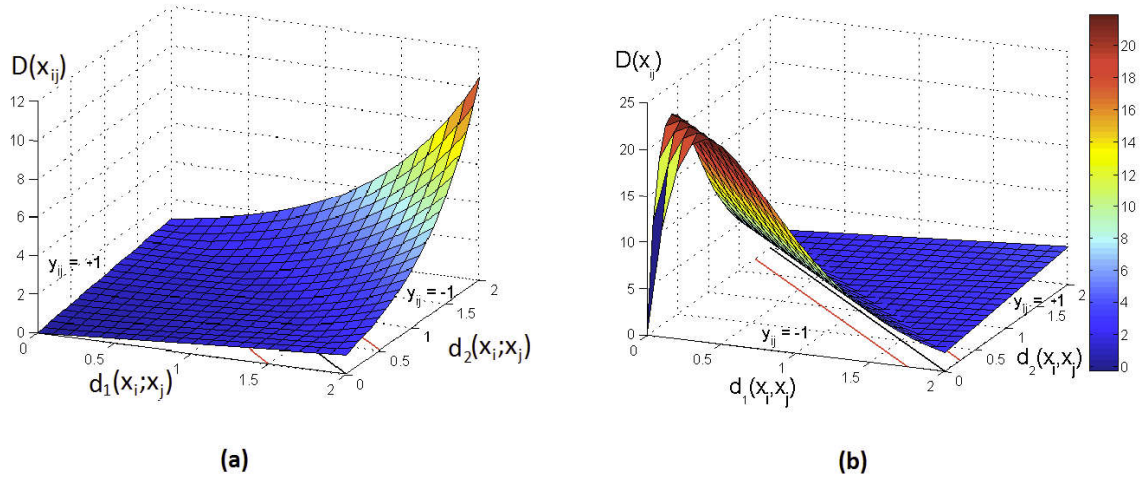

Figure 3.8: The behavior of the learned metric D ($p = 2$; $\lambda = 2.5$) with respect to common (a) and challenging (b) configurations of pull and push pairs.

Fig. 3.8, illustrates for $p = 2$ the behavior of the learned dissimilarity according to two extreme cases. The first one (Fig. 3.8-a), represents common expected configuration where pairs $Pull_i$ are situated in the same side as the origin. The dissimilarity increases proportionally to the norm in the pull side, then exponentially on the push side. Although the expansion operated in the push side is dispensable in that case, it doesn't affect nearest neighbors classification. Fig. 3.8-b, shows a challenging configuration where pairs $Push_i$ are situated in the same side as the origin. The dissimilarity behaves proportionally to the norm on the pull 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 pairs $Push_i$, *i.e.*, pairs of norm zero labeled differently.

3.6.3 Non-linearly separable Pull and Push sets

The above solution holds true for any kernel κ and allows to extend the dissimilarity D given in Eq. 3.36 to non linearly separable pull and push pairs. Let κ be a kernel defined in the dissimilarity 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:

$$D_{\mathscr{H}}(\mathbf{x}_{i,test}) = |(||\mathbf{P}_{\mathbf{w}}(\Phi(\mathbf{x}_{i,test}))|| - ||\mathbf{P}_{\mathbf{w}}(\Phi(\mathbf{0}))||)| \cdot \exp\left(\lambda \left[- \left(\sum_{ij} y_{ij}\alpha_{ij}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{i,test}) + b \right) \right]_+ \right) \quad \lambda > 0 \tag{3.37}$$

with $\Phi(\mathbf{w})$ the image of $\mathbf{w}$ into the feature space $\mathscr{H}$. Based on Eq. 3.34, substituing Eq. 3.21 back into $\mathbf{w}$, the inner product gives $< \mathbf{w}; \Phi(\mathbf{x}_{i,test}) >= \sum_{ij} y_{ij}\alpha_{ij}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{i,test})$ and the norm of $\mathbf{w}$ gives $||\mathbf{w}|| = \sqrt{\sum_{ijkl} \alpha_{ij}\alpha_{kl}y_{ij}y_{kl}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{kl})}$. Replacing back into Eq. 3.35, the norm of the orthogonal projection of $\Phi(\mathbf{x}_{i,test})$ on $\Phi(\mathbf{w})$ gives:

$$||\mathbf{P}_{\mathbf{w}}(\Phi(\mathbf{x}_{i,test}))|| = \frac{\sum_{ij} y_{ij}\alpha_{ij}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{i,test})}{\sqrt{\sum_{ijkl} \alpha_{ij}\alpha_{kl}y_{ij}y_{kl}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{kl})}} \tag{3.38}$$

Note that as $\Phi(\mathbf{0})$ doesn't meet the origin in the feature space $\mathscr{H}$, the norms in Eq. 3.37 are centered with respect to $\Phi(\mathbf{0})$. Note also that the denominator is positive since it is equal to $||\mathbf{w}||$.

Concerning the properties of D and $D_{\mathscr{H}}$, it can be shown that reflexivity and symmetry is ensured. Positivity is ensured as D is a product of two positive terms.

Note that other propositions for D and $D_{\mathscr{H}}$ are possible. For example, we could not consider the max operator and allow $\lambda \in \mathbb{R}$. Thus, for negative values of λ, the exponential term will have a pull action. Note that in both case, the dissimilarity measure can lead to extreme values, making a risk to polarize the dissimilarity. In practice, our proposed D and $D_{\mathscr{H}}$ provides suitable solutions for the considered datasets. Moreover, note that the framework to define the metric D and $D_{\mathscr{H}}$ can also be used in the linear and quadratic formalization. However, the obtained solution for D and $D_{\mathscr{H}}$ can be far away from the original form of D that was optimized in the optimization problem.

3.7 SVM-based solution and algorithm for M^{2}TML

In this section, we review the main steps of our solutions based on SVM. In particular, we detail two pre-processing steps needed to adapt the SVM framework to our metric learning problem: pairwise space normalization, neighborhood scaling.

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 as explained in Section 1.1.4.

Neighborhood scaling

In real datasets, local neighborhoods can have very different scales as illustrated in Fig. E.1. To make the pull neighborhood spreads comparable, we propose for each $\mathbf{x}_i$ to scale each pairs $\mathbf{x}_{ij}$ such that the L_2 norm (radius) of the farthest m-th nearest neighbor is 1:

$$\mathbf{x}_{ij}^{norm} = \left[\frac{d_1(\mathbf{x}_{ij})}{r_i}, \ldots, \frac{d_p(\mathbf{x}_{ij})}{r_i}\right]^T \tag{3.39}$$

where r_i is the radius associated to $\mathbf{x}_i$ corresponding to the maximum norm of its m-th nearest neighbor of same class in $Pull_i$:

$$r_i = \max_{\mathbf{x}_{ij} \in Pull_i} ||\mathbf{x}_{ij}||_2 \tag{3.40}$$

For simplification purpose, we denote $\mathbf{x}_{ij}$ as $\mathbf{x}_{ij}^{norm}$. Fig. 3.9 illustrates the effect of neighborhood scaling in the dissimilarity space.

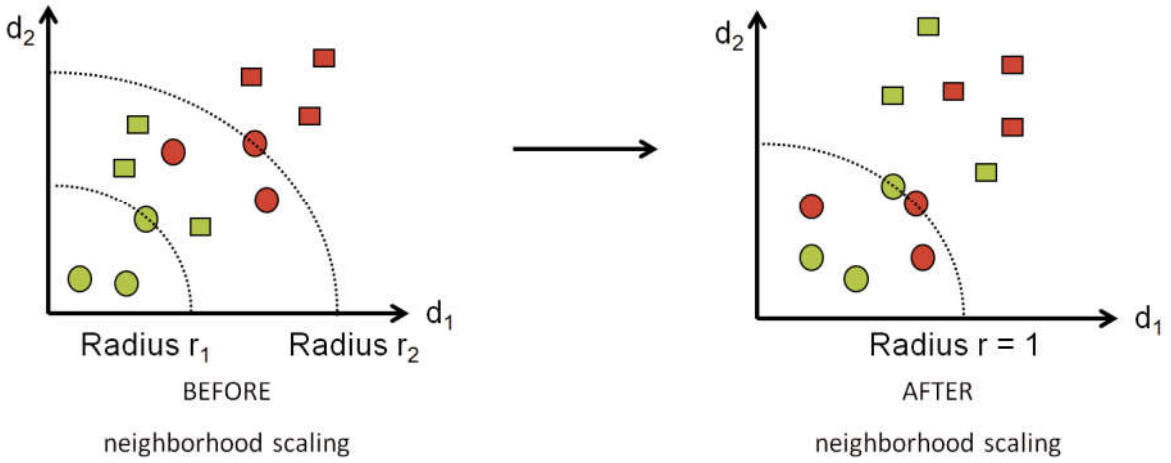

Figure 3.9: Effect of neighborhood scaling before (left) and after (right) on the neighborhood of two time series $\mathbf{x}_1$ (green) and $\mathbf{x}_2$ (red). Circle represent pull pairs $Pull_i$ and square represents push pairs $Push_i$ for $m = 3$ neighbors. Before scaling, the problem is not linearly 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.

Finally, 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.

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

1: Input: $\{\mathbf{x}_i, y_i\}_{i=1}^n$ n labeled time series
$d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 3.2
a kernel κ
2: Output: the learned dissimilarity D or $D_{\mathscr{H}}$ depending of κ
3: *Dissimilarity embedding*
Embed pairs $(\mathbf{x}_i, \mathbf{x}_j)$ $i, j \in 1, ..., n$ into $\mathscr{E}$ as described in Eq. 3.1 and normalize d_hs
4: *Build $Pull_i$ and $Push_i$ sets*
Build the sets of pairs $Pull_i$ and $Push_i$ as described in Eq. 3.5 & 3.6 and scale the radii to 1 (Eq. 3.39).
5: SVM *learning*
Train a SVM for a large margin classifier between $Pull_i$ and $Push_i$ sets (Eq. 3.33)
6: *Dissimilarity definition*
Consider Eq. 3.36 (resp. Eq. 3.37) to define D (resp. $D_{\mathscr{H}}$) a linear (resp. non linear) combination function of the metrics d_hs.

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

1: Input: $\{\mathbf{x}_i, y_i\}_{i=1}^n$ 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, 3.2
the learned dissimilarity D or $D_{\mathscr{H}}$ depending of the kernel κ
2: Output: Predicted label $\hat{y}_{test}$
3: *Dissimilarity embedding*
Embed pairs $(\mathbf{x}_i, \mathbf{x}_{test})$ $i \in 1, ..., n$ into $\mathscr{E}$ as described in Eq. 3.1 and normalize d_hs using the same normalization parameters in Algorithm 1
4: *Combined metric computation*
Consider Eq. 3.36 (resp. Eq. 3.37) 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 $d_h(\mathbf{x}_i, \mathbf{x}_{test})$.
5: *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 $\mathbf{x}_{test}$

3.8 Conclusion

To learn a multi-modal and multi-scale temporal combined metric, we propose in this chapter to embed time series into a dissimilarity space. The multi-modal and multi-scale metric learning problem can be formalized as a problem of learning a function in the dissimilarity space, that ensures the properties of a dissimilarity. We formulate the metric learning problem into a general optimization problem involving a pull and push term. Choosing a m-neighborhood, greater than the k-neighborhood allows to generalize better the learnt metric. From the general formalization, we propose three different formalizations (Linear, Quadratic, SVM-based). Table 3.1 sums up the main pros and cons of each formalization.

	Linear formalization	Quadratic formalization	SVM-based formalization
Linear solution	Yes	Yes	Yes
Non-linear solution	No	Yes	Yes
Sparcity	Yes	No	Yes/No
Form of the metric D	*a priori*	*a priori*	*built*
Positivity of D	Yes	No	Yes
Symmetry of D	Yes	Yes	Yes
Reflexivity of D	Yes	Yes	Yes

Table 3.1: The different formalizations for M^{2}TML

The adaptation of SVM in the dissimilarity space to learn the 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 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.