Chapter 3

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

Contents

In this chapter, we first motivate the problem of Multi-modal and Multi-scale Temporal Metric Learning (M^2TML) for nearest neighbors classification. Secondly, we recall the Large Margin Nearest Neighbors (LMNN) framework proposed by Weinberger & Saul. Thirdly, we introduce the concept of dissimilarity space. Then, we formalize the general problem of M^2TML. After that, 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 give the algorithm.

3.1 Motivations

This work focuses on defining a 'good' metric for classification of time series. The definition of a metric to compare samples is a fundamental issue in data analysis or machine learning. As seen in Chapter 2, temporal data may be compared based on one or several characteristics, called **modalities** (amplitude, behavior, frequency) and they might be subjected to delays. In some classification applications, the most discriminative characteristic between time series of different classes can be localized on a smaller part of the signal (scale). We believe that the definition of a temporal metric should consider at least these different aspects (modality, delay, scale) in order to improve the performance of a classifier. Fig. 3.1 illustrates a result obtained with our proposition. There is a 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 by black rectangles in the figure). The performance of the learned 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).

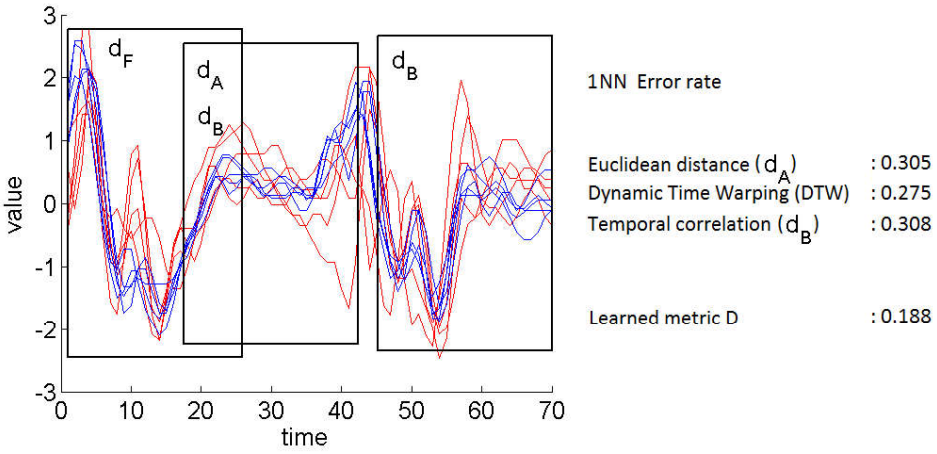

Figure 3.1: SonyAIBO dataset and error rate using a kNN ($k = 1$) with standard metrics (Euclidean distance, Dynamic Time Warping, temporal correlation) and a learned combined metric D. The figure shows the 4 major metrics involve in the combined metric D and their respective temporal scale (black rectangles).

Our aim is to take leverage from the 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 unimodal metrics at the global scale.
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) that brings closer samples that are expected to be similar, and pushes far away those expected to be dissimilar. Such similarity and dissimilarity expectations, is 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 the

last decade 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 at capturing non linear structure in the data, *e.g.*, Kernel Principal Component Analysis (KPCA) and Support Vector Metric Learning (SVML). In both cases, the metric is directly learned in the original space (*i.e.*, space described by the features of the samples). In KPCA, the aim is to project the data into a non linear feature space and learn the metric in that projected space [ZY10]; [Cha+10]. In SVML, the Mahalanobis distance is learned jointly with the learning of the SVM model in order to minimize the validation error [XWC12]. In general, the optimization problems in non linear approaches is more expensive to solve that in linear approaches, and the methods tend to favor overfitting as the constraints are generally easier to satisfy in a nonlinear kernel space. A more detailed review on metric learning is done in [BHS13].
Contrary to static data, metric learning for structured data (*e.g.* sequence, time series, trees, graphs, strings) is less frequent. 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 most of these propositions, temporal metric learning process is systematically: a) Uni-modal (amplitude-based), the divergence between aligned elements being either the Euclidean or the Mahalanobis distance and b) Uni-scale (global level), involving all time series elements at once, 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 pairwise dissimilarity space where a linear function combining several modalities at different temporal scales can be learned, driven by a large margin optimization process inspired from the 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 this chapter, we first recall the Large Margin Nearest Neighbors (LMNN) framework proposed by Weinberger & Saul. Secondly, we introduce the concept of pairwise dissimilarity space. 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 learned metrics. Finally, we give the algorithm. Note that these formalizations don't concern only time series and could be applied to learn a combined metric on any type of data.

3.2 A recall on Large Margin Nearest Neighbors (LMNN)

Let $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ be a set of n static vector samples, $\mathbf{x}_i \in \mathbb{R}^p$, p being the number of descriptive features and y_i the class labels. Weinberger & Saul proposed in [WS09a] an approach to learn a metric D for a large margin k-NN classifier in the case of static data.

Large Margin Nearest Neighbor (LMNN) approach is based on two intuitions: first, each training sample $\mathbf{x}_i$ should have the same label y_i as its k nearest neighbors; second, training samples with different labels should be widely separated. For this, the concept of **target** and **impostors** for each training sample $\mathbf{x}_i$ is introduced. Given a metric D, target neighbors of $\mathbf{x}_i$, noted $j \rightsquigarrow i$, are the k closest $\mathbf{x}_j$ of the same class ($y_j = y_i$), while impostors of $\mathbf{x}_i$, denoted, $l \nrightarrow i$, are the $\mathbf{x}_l$ of different class ($y_l \neq y_i$) that invade the perimeter defined by the farthest targets of $\mathbf{x}_i$. Mathematically, for a sample $\mathbf{x}_i$, an imposter $\mathbf{x}_l$ is defined by an inequality related to the targets $\mathbf{x}_j$: $\forall l, \exists j \in j \rightsquigarrow i/$

$$D(\mathbf{x}_i, \mathbf{x}_l) \leq D(\mathbf{x}_i, \mathbf{x}_j) + 1 \tag{3.1}$$

Geometrically, an imposter $\mathbf{x}_l$ is a sample that invades the target neighborhood plus one unit margin as illustrated in Fig. 3.2. The target neighborhood is defined with respect to an initial metric D_0. Without prior knowledge, L2-norm is often used. Metric learning by LMNN aims at minimizing the number of impostors invading the target neighborhood. By adding a margin safety of one, the model is ensured to be robust to small amounts of noise in the training sample (large margin). The learned metric D pulls the targets $\mathbf{x}_j$ and pushes the impostors $\mathbf{x}_l$ as illustrated in Fig. 3.2.

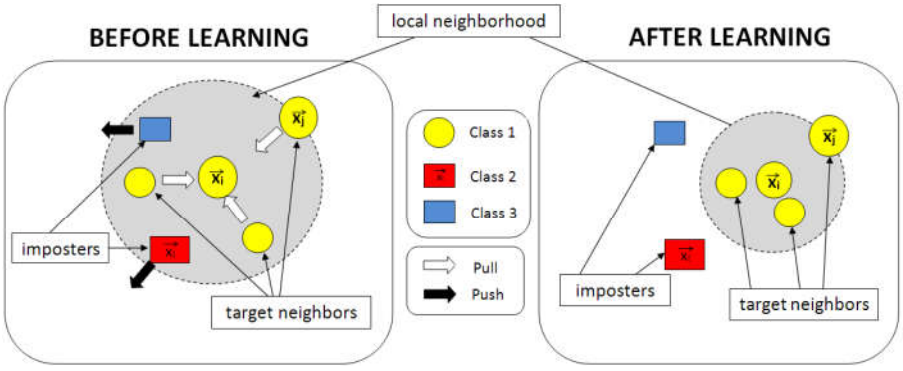

Figure 3.2: Pushed and pulled samples in the $k = 3$ target neighborhood of $\mathbf{x}_i$ before (left) and after (right) learning. The pushed (vs. pulled) samples are indicated by a white (vs. black) arrows (Weinberger & Saul [WS09a]). Note: the representation of the metric here is the one where the distance sphere is fixed and the data points are moving according to the considered distance (Section 2.2).

LMNN approach learns a Mahalanobis distance D for a robust k-NN. We recall that the k-NN decision rule will correctly classify a sample if the majority of its k nearest neighbors share the same label (Section 1.2.1). The objective of LMNN is to increase the number of samples with this property by learning a linear transformation $\mathbf{L}$ of the input space ($\mathbf{x}_i = \mathbf{L}.\mathbf{x}_i$) before

applying the k-NN classification:

$$D^2_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j) = D^2(\mathbf{L}\mathbf{x}_i, \mathbf{L}\mathbf{x}_j) \tag{3.2}$$
$$D^2_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j)||^2_2 \tag{3.3}$$

Commonly, the squared distances can be expressed in terms of a square matrix:

$$D^2_{\mathbf{L}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)\mathbf{L}^T\mathbf{L}(\mathbf{x}_i - \mathbf{x}_j) \tag{3.4}$$

Let $\mathbf{M} = \mathbf{L}'\mathbf{L}$. It is proved that any matrix $\mathbf{M}$ formed as below from a real-valued matrix $\mathbf{L}$ is positive semidefinite (*i.e.*, no negative eigenvalues) [WS09a]. Using the matrix $\mathbf{M}$, squared distances can be expressed as:

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \tag{3.5}$$

The computation of the learned metric $D_{\mathbf{M}}$ can thus be seen as a two steps procedure: first, it computes a linear transformation of the samples $\mathbf{x}_i$ given by the transformation $\mathbf{L}$; second, it computes the Euclidean distance in the transformed space:

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = D^2_{\mathbf{L}}(\mathbf{L}\mathbf{x}_i, \mathbf{L}\mathbf{x}_j) \tag{3.6}$$

Learning the linear transformation $\mathbf{L}$ is thus equivalent to learn the corresponding Mahalanobis metric D parametrized by $\mathbf{M}$. This equivalence leads to two different approaches to metric learning: we can either estimate the linear transformation $\mathbf{L}$, or estimate a positive semidefinite matrix $\mathbf{M}$. LMNN solution refers on the latter one.

Mathematically, the metric learning problem can be formalized as an optimization problem involving two terms for each sample $\mathbf{x}_i$: one term penalizes large distances between nearby inputs with the same label (pull), while the other term penalizes small distances between inputs with different labels (push). For all samples $\mathbf{x}_i$, this implies a minimization problem:

$$\begin{aligned}
&\left\{ \underset{\mathbf{M},\xi}{\operatorname{argmin}} \underbrace{\sum_{i,j \rightsquigarrow i} D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)}_{pull} + C \underbrace{\sum_{i,j \rightsquigarrow i, l \nrightarrow i} \xi_{ijl}}_{push} \right\} \\
&\text{s.t. } \forall j \rightsquigarrow i, l \nrightarrow i, \\
&D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_l) - D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijl} \\
&\xi_{ijl} \geq 0 \\
&\mathbf{M} \succeq 0
\end{aligned} \tag{3.7}$$

where ξ_{ijl} are slack variables, C is a trade-off between the push and pull term and $\mathbf{M} \succeq 0$ means that $\mathbf{M}$ is a positive semidefinite matrix. Generally, the parameter C is tuned via cross validation and grid search (Section 1.1.2). Similarly to Support Vector Machine (SVM) approach, slack variables ξ_{ijl} are introduced to relax the optimization problem.

3.3 Multi-modal and multi-scale pairwise dissimilarity space

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

3.3.1 Pairwise embedding

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 **pairwise 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 pairwise dissimilarity space $\mathscr{E} = \mathbb{R}^p$ whose dimensions are $d_1, \ldots, d_p$ (Fig. 3.3):

$$\begin{aligned} \varphi : \mathbb{R}^q \times \mathbb{R}^q &\to \mathscr{E} = \mathbb{R}^p \\ (\mathbf{x}_i, \mathbf{x}_j) &\to \mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), \ldots, d_p(\mathbf{x}_i, \mathbf{x}_j)]^T \end{aligned} \tag{3.8}$$

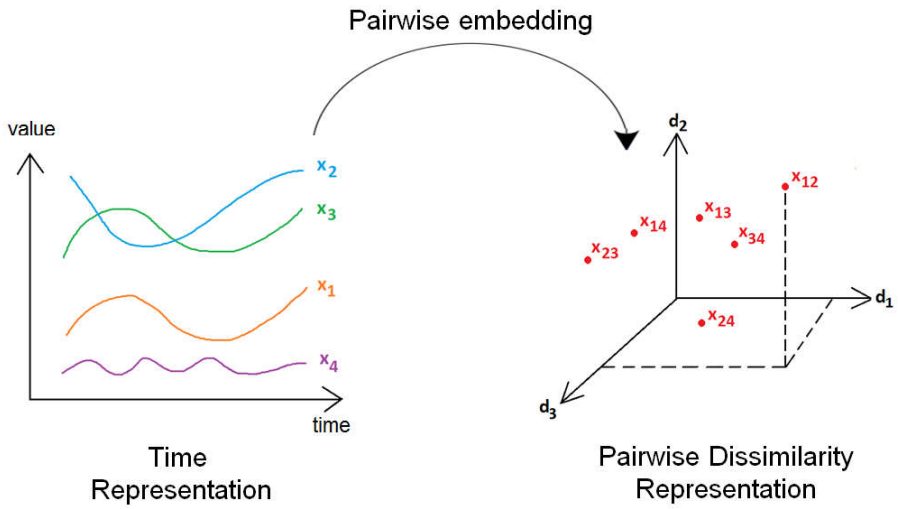

Figure 3.3: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the dissimilarity space (right) for $p = 3$ basic metrics.

A metric D that combines the p metrics $d_1, \ldots, d_p$ can be seen as a function of the dissimilarity space:

$$\begin{aligned} D : \mathbb{R}^p &\to \mathbb{R} \\ \mathbf{x}_{ij} &\to D(\mathbf{x}_{ij}) = f(d_1(\mathbf{x}_i, \mathbf{x}_j), \ldots, d_p(\mathbf{x}_i, \mathbf{x}_j)) \end{aligned} \tag{3.9}$$

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.

3.3.2 Interpretation in the pairwise dissimilarity space

In this section, we give more detailed interpretations in the dissimilarity space. We recall that the norm of a pairwise vector is given by:

$$||\mathbf{x}_{ij}|| = \sum_{h=1}^{p} d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{3.10}$$

In the following, we denote the norm $||\mathbf{x}_{ij}||$ as an initial distance in the dissimilarity space and call it D_0. Any other initial metric could have been chosen. The norm of a pairwise vector $\mathbf{x}_{ij}$ can be interpreted as a proximity measure: the lower the norm of $\mathbf{x}_{ij}$ is, the closer are the time series $\mathbf{x}_i$ and $\mathbf{x}_j$. Two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ that are on a same line that passes through the origin $\mathbf{x}_{ii} = \mathbf{0}$ represent differences in the the same proportions between their respective modalities (Fig. 3.4).

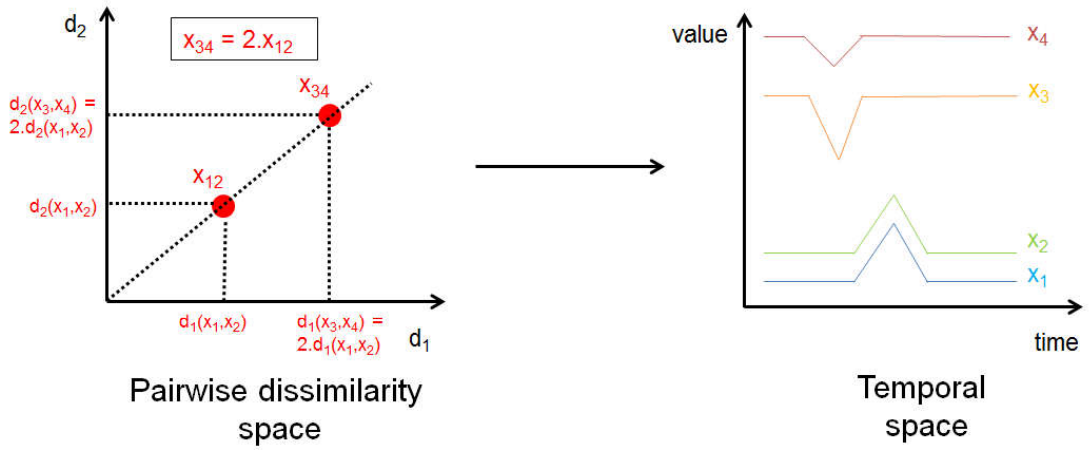

Figure 3.4: Example of interpretation of two pairwise vectors $\mathbf{x}_{12}$ and $\mathbf{x}_{34}$ on a same line passing through the origin in the pairwise dissimilarity space.

The Euclidean distance $\sqrt{\sum_{h=1}^{p} (d_h(\mathbf{x}_i, \mathbf{x}_j) - d_h(\mathbf{x}_k, \mathbf{x}_l))^2}$ between two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ represents the similarity between the differences among the same modalities, in the same proportions. Note that if the Euclidean distance is close to 0 ($\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ are close in the dissimilarity space), it doesn't mean that the time series $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ and $\mathbf{x}_l$ are similar. Fig 3.5 shows an example of two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ close together in the pairwise space. However, in the temporal space, the time series $\mathbf{x}_1$ and $\mathbf{x}_3$ are not similar for example. It means that $\mathbf{x}_i$ is as similar to $\mathbf{x}_j$ as $\mathbf{x}_k$ is to $\mathbf{x}_l$, *i.e.*, the distance D_0 between $\mathbf{x}_i$ and $\mathbf{x}_j$ is nearly the same than the distance D_0 between $\mathbf{x}_k$ and $\mathbf{x}_l$.

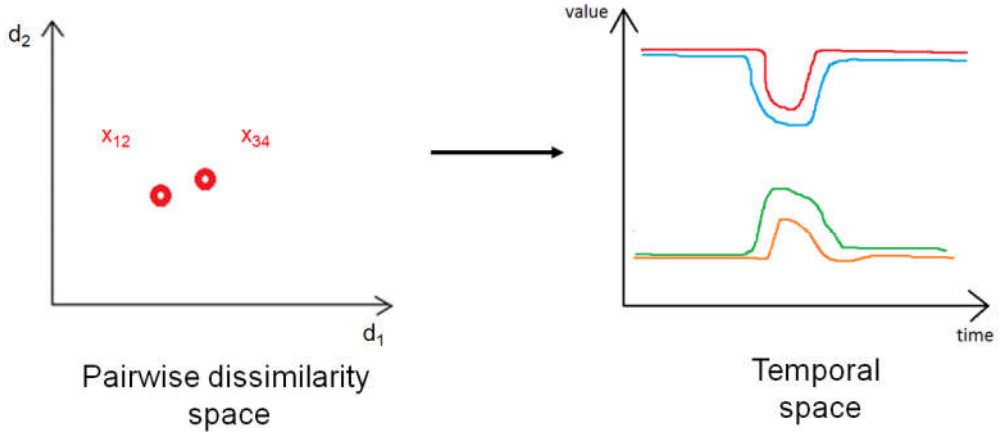

Figure 3.5: Example of two pairwise vectors $\mathbf{x}_{12}$ and $\mathbf{x}_{34}$ close in the pairwise dissimilarity space. However, the time series $\mathbf{x}_1$ and $\mathbf{x}_3$ are not similar in the temporal space.

3.3.3 Multi-scale description for time series

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 this 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 [Keo+03] or the dichotomy [DCA11]. We detail 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 local level. Many approaches have been proposed assuming fixed either the number of the segments or their lengths [Fu11]. In this 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 overlapped intervals I_L (left interval) and I_R (right interval). A parameter μ allows to overlap the two intervals I_L and I_R, covering discriminating subsequences in the central region of I (around $\frac{b+a}{2}$): $I = [a; b]; I_L = [a; a + \mu(b - a)]; I_R = [b - \mu(b - a); b]$. For $\mu = 0.6$, 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. 3.6.

A multi-scale dissimilarity description between two time series is obtained by computing the usual time series metrics (d_A, d_B, d_F) on each of 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 $d_A^{I_s}$ is $\{d_A^{I_1}, d_A^{I_2}, \ldots\}$ where $d_A^{I_s}$ is defined as:

$$d_A^{I_s}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t \in I_s} (x_{it} - x_{jt})^2} \tag{3.11}$$

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

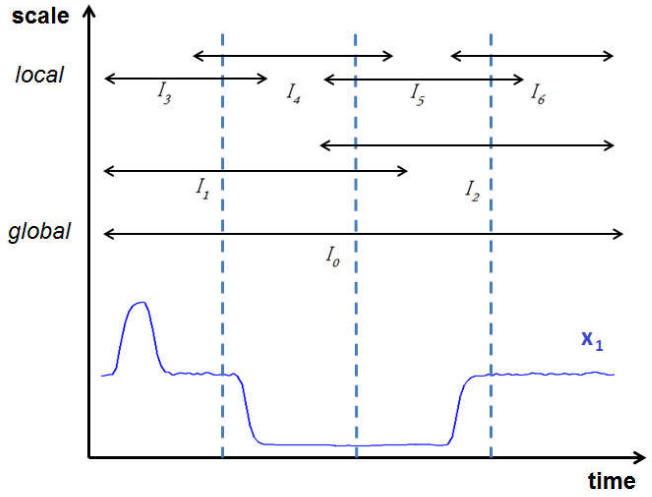

Figure 3.6: Multi-scale decomposition

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

3.4 M²TML general problem

In this section, we propose to define the Multi-modal and Multi-scale Time series Metric Learning (M²TML) problem in the initial space as a general problem of learning a function in the pairwise dissimilarity space. First, we give the intuition and formalize the general optimization problem. Secondly, we propose different strategies to define the neighborhood. Thirdly, we give some more detailed interpretations of the M²TML problem in the pairwise dissimilarity space.

3.4.1 General formalization for M²TML

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 ($D(\mathbf{x}_{ij} \geq 0)$), reflexivity ($D(\mathbf{x}_{ii}) = 0 \ \forall i$) and symmetry ($D(\mathbf{x}_{ij}) = D(\mathbf{x}_{ji}) \ \forall i, j$) (Section 2.2). In the following, the term metric is used to reference both a distance or a dissimilarity measure.

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 $Push_i$. In addition, in order to have a robust k-NN, a safety margin between the resulting metric values between the sets $Pull_i$ and $Push_i$ must be considered.

Our proposition is inspired from the LMNN framework where the optimization problem involves a pull term that penalizes large distances between sample of same labels ($Pull_i$). It can be interpreted as a regularization term on $Pull_i$. In LMNN, the push term penalizes small distances between samples of different labels ($Push_i$). It can be interpreted as a loss term on $Push_i$. To ensure a safety margin between similar and dissimilar samples, a constraint is added: $D^2(\mathbf{x}_i, \mathbf{x}_l) - D^2(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijl}$.

Similarly, we formalize the M²TML problem as an optimization problem involving both a **regularization term** on D and the pull set $Pull_i$, denoted $R_{Pull}(D)$, and a **loss term** on ξ and the push set $Push_i$, denoted $L_{Push}(\xi)$. **A set of constraints** is added to control the push term in order to have a large margin between $Pull_i$ and $Push_i$:

$$\begin{aligned} &\underset{D,\xi}{\operatorname{argmin}} \{R_{Pull}(D) + L_{Push}(\xi)\} \\ &\text{s.t. } \forall i, j \in Pull_i, l \in Push_i, \\ &\quad D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ &\quad \xi_{ijl} \geq 0 \end{aligned} \tag{3.12}$$

Among the possibilities for the regularization term, we decide to choose to minimize the sum of the distances of the pull pairs. Among the possibilities for the loss term, we decide to choose to minimize the sum of the slack variables on the push pairs:

$$R_{Pull}(D) = \sum_{\substack{i \\ j \in Pull_i}} D(\mathbf{x}_{ij}) \tag{3.13}$$

$$L_{Push}(\xi) = \sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \xi_{ijl} \tag{3.14}$$

The M²TML problem for large margin k-NN classification can be written as the following optimization problem:

$$\begin{aligned} &\underset{D,\xi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{\substack{i \\ j \in Pull_i}} D(\mathbf{x}_{ij})}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \xi_{ijl}}_{push} \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &\quad D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ &\quad \xi_{ijl} \geq 0 \end{aligned} \tag{3.15}$$

where ξ_{ijl} are the slack variables and C, the trade-off between the pull (regularization) and push (loss) costs. In the next section, we detail different strategies to define the $Pull_i$ and $Push_i$ sets.

3.4.2 Push and pull set definition

To build the pairwise training set, we associate for each $\mathbf{x}_i$, two sets, $Pull_i$ and $Push_i$, where the two sets are chosen according to one of the following strategies, illustrated in Fig 3.7. Recall that the norm $D_0(\mathbf{x}_{ij}) = ||\mathbf{x}_{ij}||_2$ is set as our initial distance D_0.

1. k-**NN vs impostors**: for a given $\mathbf{x}_i$, the sets of pairs to pull and to push corresponds respectively to:

$$\forall i \in 1, \ldots, n, \quad Pull_i = \{\mathbf{x}_{ij} \ / \ y_j = y_i, \ D_0(\mathbf{x}_{ij}) \text{ is among the } k\text{-lowest distance}\} \tag{3.16}$$

$$Push_i = \{\mathbf{x}_{il} \ / \ y_l \neq y_i, \ D_0(\mathbf{x}_{il}) \leq \max_{\mathbf{x}_{ij} \in Pull_i} D_0(\mathbf{x}_{ij})\} \tag{3.17}$$

2. k-**NN vs all**: for a given $\mathbf{x}_i$, the sets of pairs to pull and to push corresponds respectively to:

$$\forall i \in 1, \ldots, n, \quad Pull_i = \{\mathbf{x}_{ij} \ / \ y_j = y_i, \ D_0(\mathbf{x}_{ij}) \text{ is among the } k\text{-lowest distance}\} \tag{3.18}$$

$$Push_i = \{\mathbf{x}_{il} \ / \ y_l \neq y_i\} \tag{3.19}$$

3. m-**NN**$^+$ **vs** m-**NN**$^-$: for a given $\mathbf{x}_i$, the pull and push sets are defined respectively as the set of the m-nearest neighbors of the same class ($y_j = y_i$), and the m-nearest neighbor of $\mathbf{x}_i$ of a different class ($y_j \neq y_i$) More precisely, our proposition states: $m = \alpha.k$ with $\alpha \geq 1$. Other propositions for m are possible:

$$\forall i \in 1, \ldots, n, \quad Pull_i = \{\mathbf{x}_{ij} \ / \ y_j = y_i, \ D_0(\mathbf{x}_{ij}) \text{ is among the } m\text{-lowest distance}\} \tag{3.20}$$

$$Push_i = \{\mathbf{x}_{il} \ / \text{ s.t. } y_l \neq y_i, \ D_0(\mathbf{x}_{il}) \text{ is among the } m\text{-lowest distance}\} \tag{3.21}$$

In the following, we denote $m\text{-NN}^+ = \bigcup_i Pull_i$ and $m\text{-NN}^- = \bigcup_i Push_i$

Finally, let discuss about the similarities and differences between LMNN (Weinberger & Saul [WS09a]) and our M²TML proposition. In LMNN, the sets $Pull_i$ and $Push_i$ are defined according the k-**NN vs impostors** strategy (Eqs. 3.16 & 3.17) and may be unbalanced. The sets are defined and fixed during the optimization process according to the initial metric D_0. In M²TML the sets $Pull_i$ and $Push_i$ are defined according the m-**NN**$^+$ **vs** m-**NN**$^-$ strategy (Eqs. 3.20 & 3.21) and are balanced. The sets are defined and fixed during the optimization process according to the initial metric D_0, but the m-neighborhood is larger than the k-neighborhood. By considering a neighborhood larger than the k-neighborhood, we believe that the generalization properties of the learned metric D will be improved.

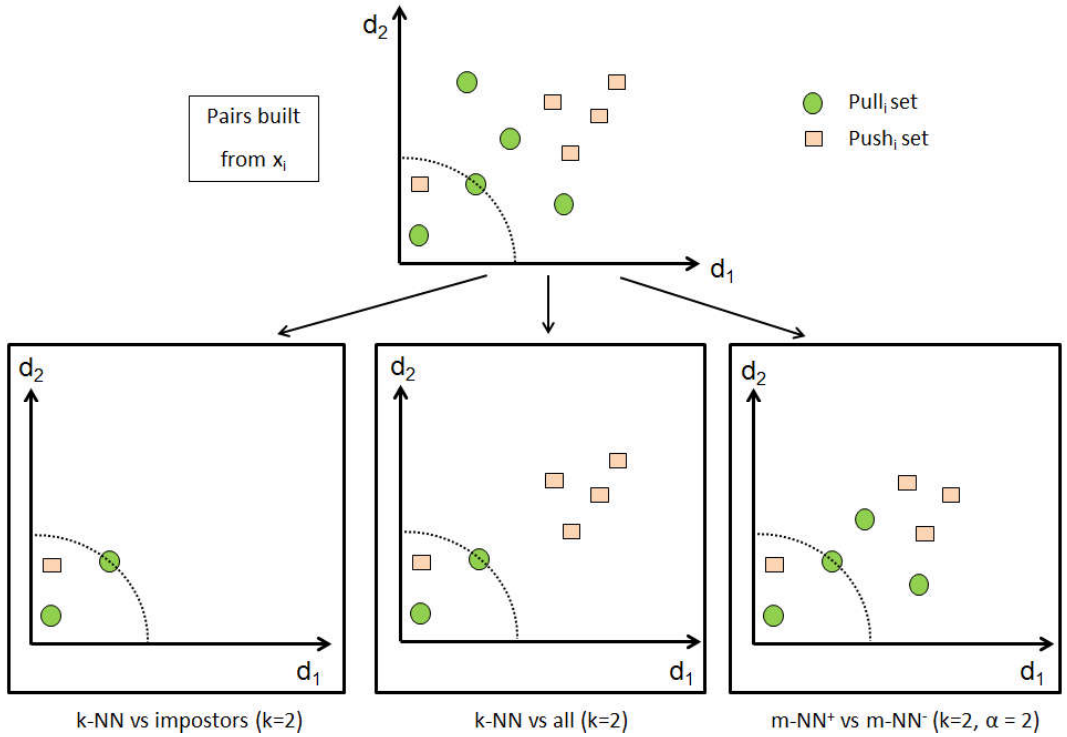

Figure 3.7: Example of different strategies to build $Pull_i$ and $Push_i$ sets for a $k = 2$ neighborhood.

3.4.3 Interpretation in the pairwise dissimilarity space

In this section, we give more detailed interpretations of the M²TML problem in the dissimilarity space. Our objective is to learn a metric D as a linear or non-linear combination of the p unimodal metrics $d_1, \ldots, d_p$. The metric D can be seen as a function of the dissimilarity space that should:

- **pull** to the origin $\mathbf{x}_{ii} = \mathbf{0}$ the pairs $\mathbf{x}_{ij}$ of $Pull_i$
- **push** away from the origin all the pairs $\mathbf{x}_{il}$ of $Push_i$

Fig. 3.8 illustrates the idea in the original space and in the pairwise dissimilarity space: first, we build the sets $Pull_i$ and $Push_i$ according to an initial metric D_0; secondly, we optimize the metric D so that the pairs $Pull_i$ are pulled to the origin and the pairs $Push_i$ are pushed away from the origin.

Note that by considering a larger neighborhood, we ensure that pairs $Push_i$ doesn't invade the perimeter defined by pairs $Pull_i$ during the optimization process. Similarly to the interpretation of slack variables in SVM, if a push pair invade the perimeter defined by pairs $Pull_i$, then in Eq. 3.15, it will violate the constraints and the slack variables ξ_{ijl} will be penalized in the objective function:

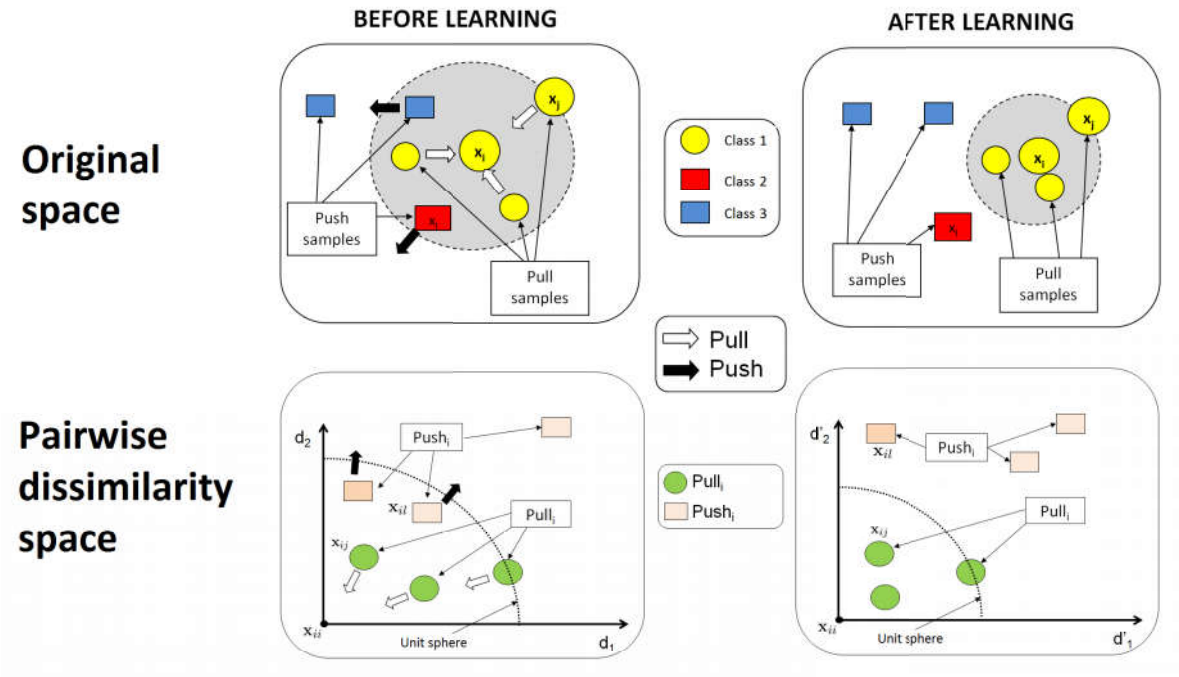

Figure 3.8: Metric learning problem in the original space (top) and the pairwise dissimilarity space (bottom) for a $k = 3$ neighborhood of $\mathbf{x}_i$. Before learning (left), push samples $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the dissimilarity pairwise space, this is equivalent to have push pairwise vectors $\mathbf{x}_{il}$ with an initial distance D_0 lower than the distance of pull pairwise vectors $\mathbf{x}_{ij}$. The aim of M²TML is to learn a metric D to push $\mathbf{x}_{il}$ (black arrow) and pull $\mathbf{x}_{ij}$ from the origin (white arrow).

- If $D(\mathbf{x}_{il}) < D(\mathbf{x}_{ij})$, then the pairs $\mathbf{x}_{il}$ is an imposter pair that invades the neighborhood of the target pairs $\mathbf{x}_{ij}$. The slack variable $\xi_{ijl} > 1$ will be penalized in the objective function.

- If $D(\mathbf{x}_{il}) \geq D(\mathbf{x}_{ij})$ but $D(\mathbf{x}_{il}) \leq D(\mathbf{x}_{ij}) + 1$, the pair $\mathbf{x}_{il}$ is within the safety margin of the target pairs $\mathbf{x}_{ij}$. The slack variable $\xi_{ijl} \in [0; 1]$ will have a small penalization effect in the objective function.

- If $D(\mathbf{x}_{il}) > D(\mathbf{x}_{ij}) + 1$, $\xi_{ijl} = 0$ and the slack variable has no effect in the objective function.

In the following, we propose different regularizers for the pull term $R_{Pull}(D)$. First, we use a linear regularization. Secondly, we use a quadratic regularization that enables to extend the approach 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, we define the combined metric D based on the SVM solution. Finally, we sum up the retained solution (SVM-based solution) and give the main steps of the algorithm.

3.5 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 the properties of the learned metric D.

Let $\{\mathbf{x}_{ij}\}_{i,j=1}^{n}$ 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 $d_1, \ldots, d_p$. We consider a linear combination of the p metrics:

$$D(\mathbf{x}_{ij}) = \mathbf{w}^T \mathbf{x}_{ij} = \sum_{h=1}^{p} w_h . d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{3.22}$$

where $\mathbf{w} = [w_1, \ldots, w_p]^T$ is the vector of weights w_h. From Eq. 3.15, by choosing $R_{Pull}(D) = R_{Pull}(\mathbf{w}) = \sum_{i,j \in Pull_i} \mathbf{w}^T \mathbf{x}_{ij}$, learning a linear combined metric D can be formalized as follow:

$$\begin{aligned}
&\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{\substack{i \\ j \in Pull_i}} \mathbf{w}^T \mathbf{x}_{ij}}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \xi_{ijl}}_{push} \right\} \\
&\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\
&\quad \mathbf{w}^T (\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\
&\quad \xi_{ijl} \geq 0
\end{aligned} \tag{3.23}$$

$$\forall h = 1, \ldots, p, \quad w_h \geq 0 \tag{3.24}$$

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. 3.20 & 3.21.

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:

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

Concerning the properties of D, positivity is ensured with the constraints $w_h \geq 0$ (Eq. 3.24) 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.

3.6 Quadratic formalization for M²TML

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 learned metric D. Finally, we study a link between SVM and the quadratic formalization.

3.6.1 Primal and dual formalization

The formulation in Eq. 3.23 supposes that the metric D is a linear combination of the metrics d_h. The linear formalization being similar to the one of a L1-regularized SVM, it can be derived into a dual form involving only dot-products to extend the method to find non-linear solutions for D. For that, we propose to change the linear regularizer $R_{Pull}(\mathbf{w})$ in the objective function of Eq. 3.23 into a quadratic regularizer. Two solutions for $R_{Pull}(\mathbf{w})$ are studied:

$$1. \quad R_{Pull}(\mathbf{w}) = \frac{1}{2}\sum_{h=1}^{p} \sum_{\substack{i \\ j \in Pull_i}} \left(w_h d_h(\mathbf{x}_{ij})\right)^2 \tag{3.25}$$

$$2. \quad R_{Pull}(\mathbf{w}) = \frac{1}{2}\sum_{h=1}^{p} \left(\sum_{\substack{i \\ j \in Pull_i}} w_h d_h(\mathbf{x}_{ij}) \right)^2 = \frac{1}{2} m.n \sum_{h=1}^{p} \left(w_h \bar{d_h}\right)^2 \tag{3.26}$$

where $\bar{d_h} = \frac{1}{mn} \sum\limits_{\substack{i \\ j \in Pull_i}} d_h(\mathbf{x}_{ij})$ denotes the mean of the distances $d_h(\mathbf{x}_{ij})$ for each metric d_h.

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

$$R(Pull) = \frac{1}{2}\mathbf{w}^T \mathbf{M} \mathbf{w} \tag{3.27}$$

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

$$1.\quad \mathbf{M} = Diag(\mathbf{X}_{pull}^T \mathbf{X}_{pull}) = \begin{bmatrix} \sum\limits_{\substack{i \\ j \in Pull_i}} d_1^2(\mathbf{x}_{ij}) & & 0 \\ & \ddots & \\ 0 & & \sum\limits_{\substack{i \\ j \in Pull_i}} d_p^2(\mathbf{x}_{ij}) \end{bmatrix} \tag{3.28}$$

$$2.\quad \mathbf{M} = Diag(\bar{\mathbf{x}}) Diag(\bar{\mathbf{x}}) = \begin{bmatrix} \bar{d_1}^2 & & 0 \\ & \ddots & \\ 0 & & \bar{d_p}^2 \end{bmatrix} \tag{3.29}$$

where $\mathbf{X}_{pull} = \bigcup_i Pull_i$ be a $(m.n) \times p$ matrix containing the vector $\mathbf{x}_{ij} \in Pull_i$ and $\bar{\mathbf{x}} = [\bar{d_1}, \ldots, \bar{d_p}]^T$ is a vector of size p containing the mean of the metrics $\bar{d_1}, \ldots, \bar{d_p}$.

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

$$\begin{aligned} \underset{\mathbf{w},\xi}{\operatorname{argmin}} & \left\{ \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w}}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \xi_{ijl}}_{push} \right\} \\ \text{s.t. } & \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ & \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \end{aligned} \tag{3.30}$$

Note that in this case, the constraint $w_h \geq 0$ (Eq. 3.24) is not considered because the following development would not allow to obtain a formulation with only dot-product. Similarly to SVM, the formulation in Eq. 3.30 can be reduced to the maximization 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 $\mathbf{r}$:

$$\begin{aligned} L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) = & \frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w} + C\sum_{ijl} \xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\ & - \sum_{ijl} \alpha_{ijl} \left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl}) \right) \end{aligned} \tag{3.31}$$

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

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

The matrix $\mathbf{M}$ being diagonal in both case (Eqs. 3.28 & 3.29), it is thus invertible. The equations lead to:

$$\mathbf{w} = \mathbf{M}^{-1} \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.32}$$
$$r_{ijl} = C - \alpha_{ijl} \tag{3.33}$$

Substituting Eq. 3.32 and 3.33 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.31, we get the dual formulation[1]:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1} (\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &0 \leq \alpha_{ijl} \leq C \end{aligned} \tag{3.34}$$

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

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1}}_{\mathbf{w}^T} \mathbf{x}_{i'j'} \tag{3.35}$$

By developing Eq. 3.34, the dual formulation is equivalent to:

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'} \left(\mathbf{x}_{il}^T \mathbf{M}^{-1} \mathbf{x}_{i'l'} - 2\mathbf{x}_{ij}^T \mathbf{M}^{-1} \mathbf{x}_{i'l'} + \mathbf{x}_{ij}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'} \right) \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &0 \leq \alpha_{ijl} \leq C \end{aligned} \tag{3.36}$$

And the metric in Eq. 3.35 writes:

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{il}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Push \text{ set}} - \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{ij}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Pull \text{ set}} \tag{3.37}$$

[1] details of the development can be found in Appendix A

3.6.2 Non-linear combined metric

The above formula (Eqs. 3.36 and 3.37) can be extended to find non-linear function for the metric D. As $\mathbf{M}^{-1}$ is a diagonal matrix, it is invertible 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 $\mathbf{M}^{-\frac{1}{2}}$:

$$1.\quad \mathbf{M}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{\sqrt{\sum\limits_{i \atop j \in Pull_i} d_1^2(\mathbf{x}_{ij})}} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sqrt{\sum\limits_{i \atop j \in Pull_i} d_p^2(\mathbf{x}_{ij})}} \end{bmatrix} \tag{3.38}$$

$$2.\quad \mathbf{M}^{-\frac{1}{2}} = \begin{bmatrix} \frac{1}{d_1} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{d_p} \end{bmatrix} \tag{3.39}$$

Then, the formulation in Eqs. 3.36 and 3.37 can be written to involve only an inner product between pairs:

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

The inner product can be easily kernelized using the "kernel" trick:

$$< \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'} > = \kappa(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'})$$

The matrix $\mathbf{M}^{-\frac{1}{2}}$ can be thus 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. 3.36, the kernelized dual formulation becomes:

$$\begin{aligned} \underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl} \alpha_{i'j'l'} (\kappa(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'l'}) - 2\kappa(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'l'}) \\ + \kappa(\mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'})) \end{aligned} \tag{3.40}$$

s.t. $\forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i$,

$0 \leq \alpha_{ijl} \leq C$

By replacing the inner product by a kernel back into Eq. 3.37, we obtain:

$$D(\mathbf{x}_{i'j'}) = \overbrace{\sum_{ijl} \alpha_{ijl} \kappa(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Push \text{ set}} - \overbrace{\sum_{ijl} \alpha_{ijl} \kappa(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Pull \text{ set}} \tag{3.41}$$

Let's give some interpretation and discussion about the properties of D. Similarly to SVM, from Eq. 3.35, 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. 3.37 & 3.41): 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 dissimilar when 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 similar when 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.

3.6.3 Link between SVM and the quadratic formalization

Parallels between Large Margin Nearest Neighbors (LMNN) and SVM have been studied in the literature [Do+12]. 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 (Section 1.2.2).

Similarly, we study in this section a link between the quadratic formalization of M^{2}TML and a SVM problem when the form of the metric D is defined *a priori*. For that, let consider the following SVM problem:

$$\begin{aligned} &\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i \text{ or} \\ j \in Push_i}} p_i \xi_{ij} \right\} \\ &\text{s.t. } \forall i, j \in Pull_i \text{ or } j \in Push_i : \\ &y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\xi_{ij} \geq 0 \end{aligned} \tag{3.42}$$

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 $Push_i$:

$$\begin{aligned}
&\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i}} p_i^+ \xi_{ij} + C \sum_{\substack{i \\ l \in Push_i}} p_i^- \xi_{il} \right\} \\
&\text{s.t. :} \\
&\forall i,j \in Pull_i : y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\
&\forall i,l \in Push_i : y_{il}(\mathbf{w}^T \mathbf{x}_{il} + b) \geq 1 - \xi_{il} \\
&\forall i,j \in Pull_i : \xi_{ij} \geq 0 \\
&\forall i,l \in Push_i : \xi_{il} \geq 0
\end{aligned} \tag{3.43}$$

where p_i^+ and p_i^- are the weight factors for pull pairs $Pull_i$ and push pairs $Push_i$.

We show in Appendix B that solving the SVM problem in Eq. 3.43 for $\mathbf{w}$ and b solves a similar problem with a quadratic regularization in Eq. 3.30 for $D(\mathbf{x}_{ij}) = -\frac{1}{2}(\mathbf{w}^T \mathbf{x}_{ij} + b)$ and where p_i^- and p_i^+ are defined as:

$$p_i^- = \frac{Card(Pull_i)}{2} = \sum_{j \in Pull_i} \frac{1}{2} \tag{3.44}$$

$$p_i^+ = \frac{Card(Push_i)}{2} = \sum_{l \in Push_i} \frac{1}{2} \tag{3.45}$$

where $Card(Pull_i)$ and $Card(Push_i)$ denotes respectively the cardinal of the set $Pull_i$ and $Push_i$ (equal to m in the m-**NN**$^+$ **vs** m-**NN**$^-$ strategy). p_i^- can be interpreted as the half of the number pairs in $Pull_i$ and p_i^+ as the half of the number of time series in $Push_i$. Let define $\xi_{ijl} = \frac{\xi_{ij}+\xi_{il}}{2}$ and $\xi_{ijkl} = \frac{\xi_{ij}+\xi_{kl}}{2}$.

Let's underline below the main similarities and differences between the SVM problem in Eq. 3.43 and the quadratic formalization of M²TML in Eq. 3.30:

- Both problems suppose at first a linear combination for D.
- Both problems can be extended to learn non-linear combinations for D thanks to the kernel trick.
- The two problems involve different regularization terms: in the quadratic formalization, the regularizer involves a pull action (Eqs. 3.25 & 3.26), not present in SVM.
- Concerning the constraints and the slack variables:
  - 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 don't belong to the same neighborhood:

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

Geometrically, the global SVM margin includes both local neighborhoods and "inter-neighborhood" between pull and push pairs of different neighborhood.

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

Concerning the properties of the metric D, positivity is not ensured in the primal and dual formulation as there are no constraint on $\mathbf{w}$. Symmetry and reflexivity is ensured.

3.7 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.

3.7.1 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 $\mathscr{E}$:

$$\begin{aligned} &\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i,j}\xi_{ij} \\ &\textbf{s.t. } y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\xi_{ij} \geq 0 \end{aligned} \tag{3.46}$$

In the linear case, a L_1 regularization in Eq. 3.46 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. 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). This point will be detailed in Section 3.8.
Note that any multi-class problem is transformed in the dissimilarity space as a binary classification problem.

3.7.2 Solution for the 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 review in this section different interpretations in the dissimilarity space.

M²TML metric definition

Given a test pair $\mathbf{x}_{i,test}$, the norm of the pair allows to estimate the proximity between $\mathbf{x}_i$ and $\mathbf{x}_{test}$. In particular, for M²TML, two quantities are used to define the dissimilarity measure: the projected norm and the distance to the margin.

Let denote $\mathbf{P_w}(\mathbf{x}_{i,test})$, the orthogonal projection of $\mathbf{x}_{i,test}$ on the axis of direction $\mathbf{w}$:

$$\mathbf{P_w}(\mathbf{x}_{i,test}) = \frac{< \mathbf{w}, \mathbf{x}_{i,test} >}{||\mathbf{w}||^2}\mathbf{w} = \frac{\mathbf{w}^T\mathbf{x}_{i,test}}{||\mathbf{w}||^2}\mathbf{w} \tag{3.47}$$

The projected norm $||\mathbf{P_w}(\mathbf{x}_{i,test})||$ of $\mathbf{x}_{i,test}$ on the direction $\mathbf{w}$ limits the comparison of $\mathbf{x}_i$ and $\mathbf{x}_{test}$ to the features separating pull and push sets (Fig. 3.9), it is defined as:

$$||\mathbf{P_w}(\mathbf{x}_{i,test})|| = \frac{|\mathbf{w}^T\mathbf{x}_{i,test}|}{||\mathbf{w}||} \tag{3.48}$$

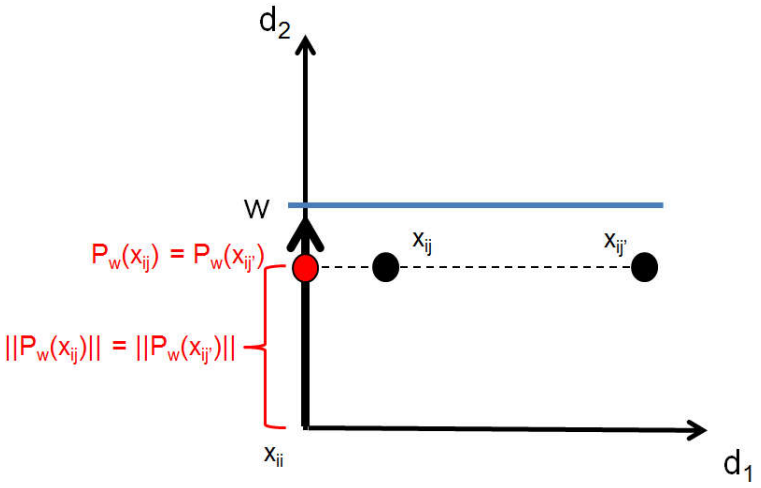

Figure 3.9: The projected vector $\mathbf{P_w}(\mathbf{x}_{ij})$ and $\mathbf{P_w}(\mathbf{x}_{ij'})$

Although the norm $||\mathbf{P_w}(\mathbf{x}_{i,test})||$ satisfies positivity, it doesn't guarantee lower distances for pull pairs than for push pairs as illustrated in Fig 3.10.

Note that the distance of the projection to the margin $\mathbf{w}^T\mathbf{P_w}(\mathbf{x}_{i,test}) + b$ gives the membership of the projected vector $\mathbf{P_w}(\mathbf{x}_{i,test})$ in the pull or push side. However, it can't used as a dissimilarity (non-positivity).

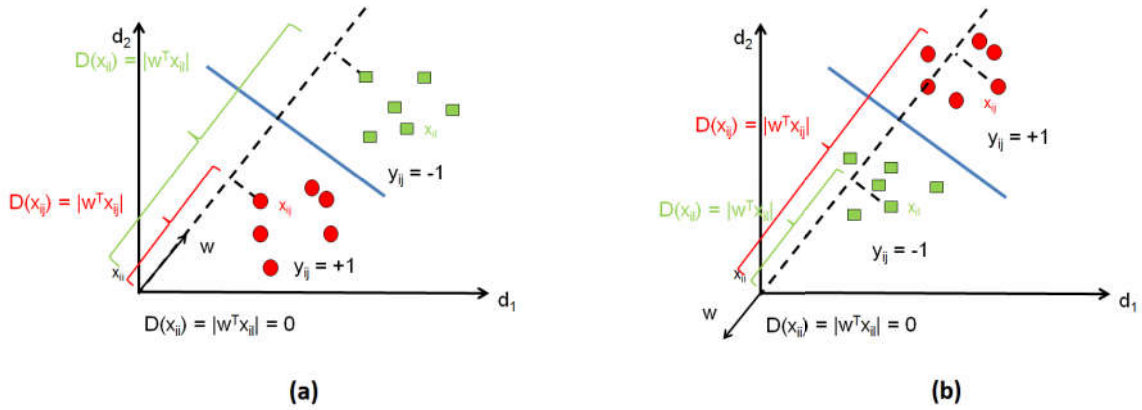

Figure 3.10: Example of SVM solutions and of the resulting metric D defined by the norm of the projection on $\mathbf{w}$. Fig. (a) represents common expected configuration where pull pairs $Pull_i$ are situated in the same side as the origin $\mathbf{x}_{ii} = \mathbf{0}$. In Fig. (b), the vector $\mathbf{w} = [-1 - 1]^T$ 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 $D(\mathbf{x}_{ij})$.

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:

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})||.\exp(\lambda[-(\mathbf{w}^T\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test}) + b)]_+) \quad \lambda > 0 \tag{3.49}$$

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

Fig. 3.11, illustrates for $p = 2$ the behavior of the learned dissimilarity according to two extreme cases. The first one (Fig. 3.11-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. 3.11-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.

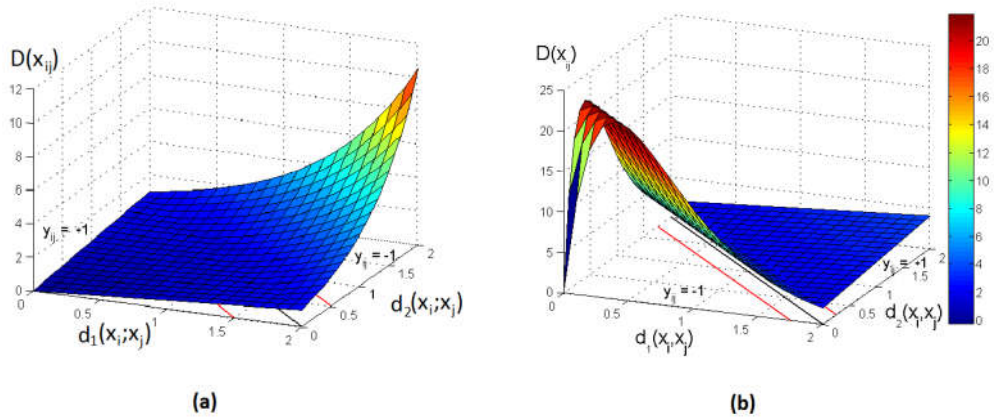

Figure 3.11: The behavior of the learned metric D ($p = 2$; $\lambda = 2.5$) with respect to common (a) and challenging (b) configurations of pull and push pairs.

3.7.3 Solution for the 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. 3.49 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:

$$D_{\mathscr{H}}(\mathbf{x}_{i,test}) = |(||\mathbf{P_w}(\Phi(\mathbf{x}_{i,test}))|| - ||\mathbf{P_w}(\Phi(\mathbf{0}))||)| \cdot \exp\left(\lambda \left[- \left(\sum_{ij} y_{ij}\alpha_{ij}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{i,test}) + b \right) \right]_+ \right) \quad \lambda > 0 \tag{3.50}$$

with $\Phi(\mathbf{x}_{i,test})$ and $\Phi(\mathbf{0})$ denotes the image of $\mathbf{x}_{i,test}$ and $\mathbf{0}$ into the feature space $\mathscr{H}$. Based on Eq. 3.47, from the known SVM equations (Section 1.2.2), 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. 3.48, the norm of the orthogonal projection of $\Phi(\mathbf{x}_{i,test})$ on $\mathbf{w}$ gives:

$$||\mathbf{P_w}(\Phi(\mathbf{x}_{i,test}))|| = \frac{\sum_{ij} y_{ij}\alpha_{ij}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{i,test})}{\sqrt{\sum_{ijkl} \alpha_{ij}\alpha_{kl}y_{ij}y_{kl}\kappa(\mathbf{x}_{ij}, \mathbf{x}_{kl})}} \tag{3.51}$$

Note that as $\Phi(\mathbf{0})$ is not guarantee to be the origin in the feature space $\mathscr{H}$, the norms in Eq. 3.50 are centered with respect to $\Phi(\mathbf{0})$ to ensure the reflexivity property. It is easy to show that both D and $D_{\mathscr{H}}$ ensure the properties of a dissimilarity (positivity, reflexivity, symmetry).

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.

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

In this section, we review the main steps of the retained SVM solution. In particular, we detail two pre-processing steps needed to adapt the SVM framework to our metric learning problem that are the pairwise space normalization and the 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 2.5.2.

Neighborhood scaling. In real datasets, local neighborhoods may have very different scales as illustrated in Fig. 3.12. 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:

$$\mathbf{x}_{ij}^{norm} = \left[\frac{d_1(\mathbf{x}_i, \mathbf{x}_j)}{r_i}, \ldots, \frac{d_p(\mathbf{x}_i, \mathbf{x}_j)}{r_i}\right]^T \tag{3.52}$$

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 $Pull_i$:

$$r_i = \max_{\mathbf{x}_{ij} \in Pull_i} D_0(\mathbf{x}_{ij}) \tag{3.53}$$

For simplification purpose, we denote $\mathbf{x}_{ij}$ as $\mathbf{x}_{ij}^{norm}$. Fig. 3.12 illustrates the effect of neighborhood scaling in the dissimilarity space.

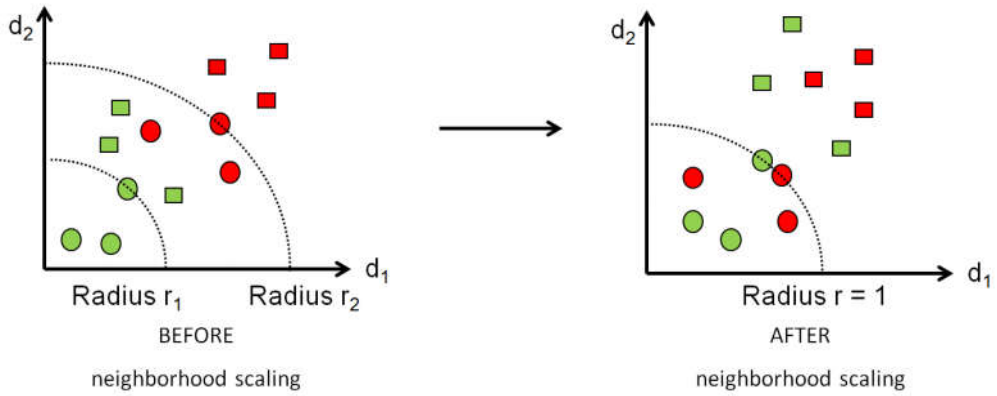

Figure 3.12: Effect of neighborhood scaling before (left) and after (right) on the neighborhood of two time series $\mathbf{x}_1$ (green) and $\mathbf{x}_2$ (red). Circle represent pairs $Pull_i$ and square represents pairs $Push_i$ for $m = 3$ neighbors. Before scaling, the problem is not linearly separable with a global SVM approach and the spread of each neighborhood are not comparable. After scaling, the target neighborhood becomes comparable and in this example, the problem becomes linearly separable.

Finally, Algorithm 1 summarizes the main steps to learn a multi-modal and multi-scale temporal metric D for a robust nearest neighbors classifier of time series. 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 of time series

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, 3.11
a kernel κ
2: Output: the learned dissimilarity D or $D_{\mathscr{H}}$ depending of κ
3: *Pairwise dissimilarity embedding and normalization*
Embed pairs $(\mathbf{x}_i, \mathbf{x}_j)$ $i, j \in 1, ..., n$ into $\mathscr{E}$ as described in Eq. 3.8 and normalize d_hs (Section 2.5.2)
4: *Build $Pull_i$ and $Push_i$ sets and neighborhood scaling*
Build the sets of pairs $Pull_i$ and $Push_i$ as described in Eq. 3.20 & 3.21 and scale the radii to 1 (Eq. 3.52).
5: SVM *learning*
Train a SVM for a large margin classifier between $Pull_i$ and $Push_i$ sets (Eq. 3.46)
6: *Dissimilarity definition*
Consider Eq. 3.49 (resp. Eq. 3.50) to define D (resp. $D_{\mathscr{H}}$) a linear (resp. non linear) combination function of the normalized 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
$\mathbf{x}_{test}$ a time series to test
$d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 3.11
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. 3.8 and normalize d_hs using the same normalization parameters than Algorithm 1
4: *Combined metric computation*
Consider Eq. 3.49 (resp. Eq. 3.50) 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 $\mathbf{x}_{test}$

3.9 Conclusion of the chapter

To learn a multi-modal and multi-scale temporal combined metric, we propose in this chapter to embed time series into a pairwise dissimilarity space. The multi-modal and multi-scale metric learning (M^{2}TML) problem can be formalized as a problem of learning a function in the pairwise dissimilarity space, that ensures the properties of a dissimilarity.
To learn a metric for a robust k-NN, we formulate the M^{2}TML problem into a general regularized large margin optimization problem involving a regularization (pull) and loss (push) term. Choosing a m-neighborhood, greater than the k-neighborhood allows to generalize better the learned metric. From the general formalization, we propose three different formalizations (Linear, Quadratic, SVM-based). Table 3.1 sums up the characteristics of each formalization and the induced dissimilarities.

	Linear formalization	Quadratic formalization	SVM-based formalization
D	Linear	Linear/Non-linear	Linear/Non-linear
Sparcity	Yes	No	Yes/No
Dissimilarity properties	Yes	No (non-positivity)	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.

As we have defined all functions components of our algorithms (learning, testing), we test our proposed algorithms M^{2}TML in the next chapter on large public datasets.