CHAPTER 3

Learning a multi-modal and multi-scale combined metric

Sommaire

In this chapter, we first motivate the problem of learning a metric that combines several metrics at different scales for a robust k-NN classifier. 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.

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).

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],

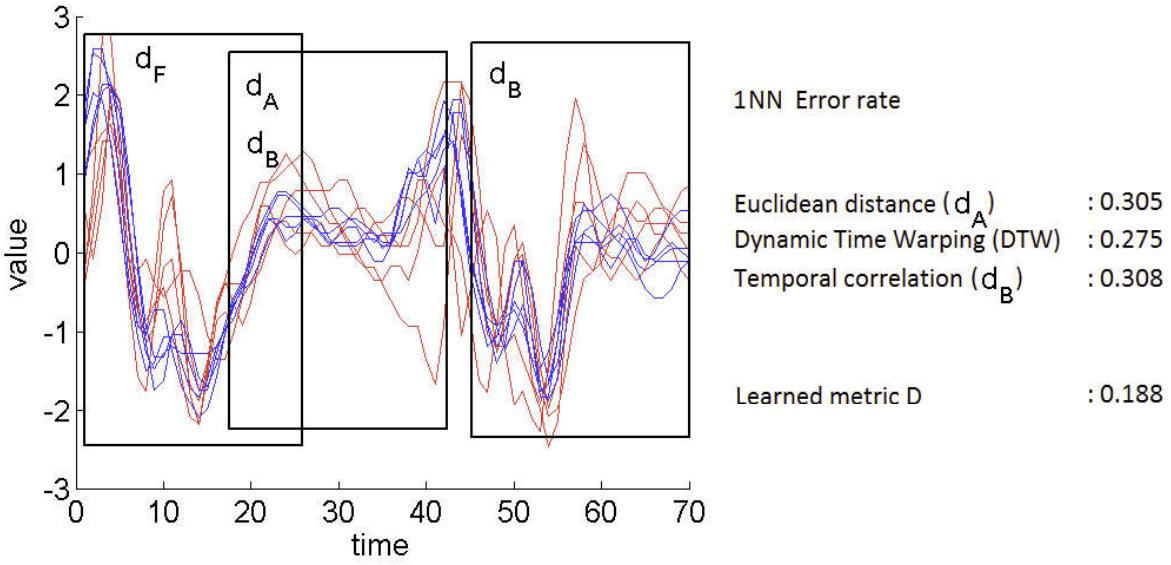

Figure 3.1: SonyAIBO dataset and error rate using a kNN ($k = 1$) with standard metrics (Euclidean distance, Dynamic Time Warping, temporal correlation) and a learned combined metric D. The figure shows the 4 major metrics involves in the combined metric D and their temporal scale (black rectangles).

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) Uni-modal (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 frame-

work [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), give an interpretation of the solutions and study the properties of the leanrt 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 Dissimilarity and multi-scale 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 $\mathbf{x}_i$. For instance, in Chapter 2, we have proposed three types of metrics for time series: amplitude-based d_A, behavior-based d_B and frequential-based d_F. Our objective is to learn a metric $D = f(d_1, \ldots, d_p)$ that combines the p 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 ψ an embedding function that maps each pair of time series $(\mathbf{x}_i; \mathbf{x}_j)$ to a vector $\mathbf{x}_{ij}$ in a dissimilarity space $\mathbb{R}^p$ whose dimensions are the dissimilarities $d_1, \ldots, d_p$ (Fig. 3.2):

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

A metric D that combines the p metrics $d_1, \ldots, d_p$ can be seen as a function of the dissimilarity space. 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 a standard Euclidean distance between two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ in the dissimilarity space cannot give any interpretation about the proximity of the time series $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ and $\mathbf{x}_l$ in the original space.

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

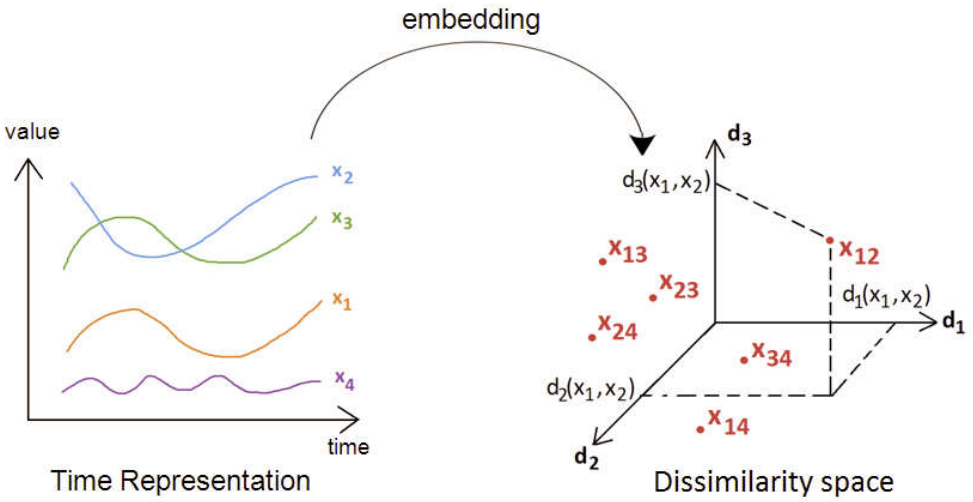

Figure 3.2: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the dissimilarity space (right) for $p = 3$ basic metrics.

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 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.

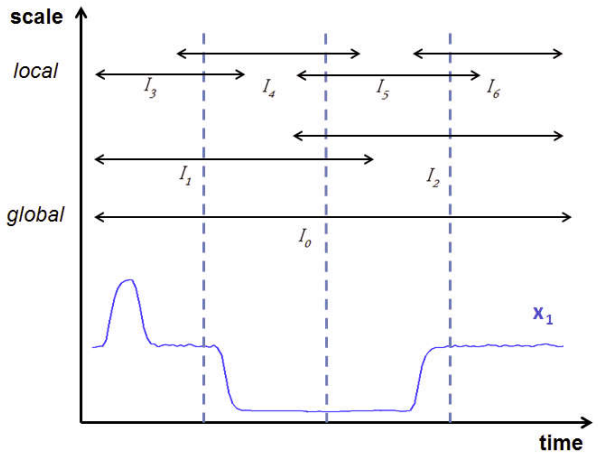

Figure 3.3: Multi-scale decomposition

3.3 Combined metric learning problem framework

In this section, we formalize the general problem of learning a combined metric. We propose to define the Multimodal and Multiscale Time series Metric Learning (M^{2}TML) problem in the initial space as a problem of learning a function in the dissimilarity space under a set of constraints. First, we give the intuition and formalize the general optimization problem. Secondly, we propose different strategies to define the neighborhood. Finally, we explain the differences between Weinberger & Saul's framework and our proposition M^{2}TML.

3.3.1 General formalization

Our objective is to learn a metric $D = f(d_1, \ldots, d_p)$ that combines the p metrics $d_1, \ldots, d_p$ for a robust k-NN classifier. The function f can be linear or non-linear and must satisfy the properties of a metric, *i.e.*, positivity, symmetry, distinguishability and triangular inequality. 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 the concept. Formally, the metric learning problem can be written as an optimization problem that involves both a regularized $R(Pull_i)$ and a loss term $L(Push_i)$ under some constraints:

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

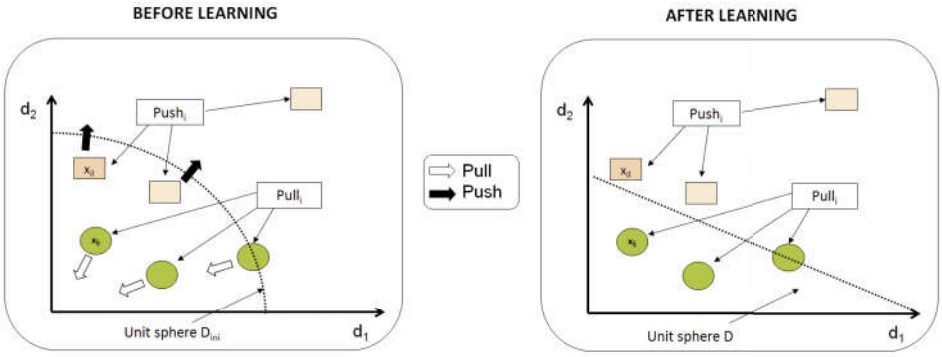

Figure 3.4: Geometric representation of the metric learning problem in the dissimilarity space for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), undesired samples $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the dissimilarity space, this is equivalent to have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

In the case of learning a combined metric D, the problem can be written as:

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

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 sets $Pull_i$ and $Push_i$.

3.3.2 Push and pull set definition

For each time series $\mathbf{x}_i$, we denote X_i^+ the set of pairs $\mathbf{x}_{ij}$ such that $y_{ij} = +1$ ($y_j \neq y_i$) and X_i^- the set of pairs $\mathbf{x}_{ij}$ such that $y_{ij} = -1$ ($y_j = y_i$):

$$X_i^- = \{\mathbf{x}_{ij}, y_{ij} = -1\} \quad \text{(same class)} \tag{3.5}$$

$$X_i^+ = \{\mathbf{x}_{ij}, y_{ij} = +1\} \quad \text{(different classes)} \tag{3.6}$$

To define the sets $Pull_i$ and $Push_i$, we use an Euclidean distance as an initial metric:

$$||\mathbf{x}_{ij}||_2 = \sqrt{\sum_{h=1}^{p} (d_h(\mathbf{x}_i, \mathbf{x}_j))^2} \tag{3.7}$$

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

$$X_i^{-*} = \{\mathbf{x}_{ij}, y_{ij} = -1\} \quad \text{s.t. } j \rightsquigarrow i \tag{3.8}$$

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

$$X_i^{+*} = \{\mathbf{x}_{il}, y_{il} = +1\} \quad \text{s.t. } l \nrightarrow i \tag{3.9}$$

To build the pairwise training set X ($Pull_i + Push_i$), we propose 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 target set and impostor set:
$$X = \bigcup_i (X_i^{-*} \cup X_i^{+*}) \tag{3.10}$$

2. k-**NN vs all**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and positive set. It ensures that no pairs $\mathbf{x}_{il}$ of different classes will invade the target neighborhood during the learning process:
$$X = \bigcup_i (X_i^{-*} \cup X_i^{+}) \tag{3.11}$$

3. m-**NN**$^+$ **vs** m-**NN**$^-$: it corresponds to the union for all $\mathbf{x}_i$ of the set of the m-nearest neighbors of the same class, denoted m-NN$_i^-$, and the m-nearest neighbor of $\mathbf{x}_i$ of a different class ($y_j \neq y_i$), denoted m-NN$_i^+$. More precisely, our proposition states: $m = \alpha.k$ with $\alpha \geq 1$. Other propositions for m are also possible:
$$X = \bigcup_i (m\text{-NN}_i^+ \cup m\text{-NN}_i^-) \tag{3.12}$$

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

In our proposition, we choose the m-**NN**$^+$ **vs** m-**NN**$^-$ strategy. By considering a neighborhood larger than the k-neighborhood (k-**NN vs impostors** strategy) during the training phase, we believe that the generalization properties of the learnt metric D would be improved

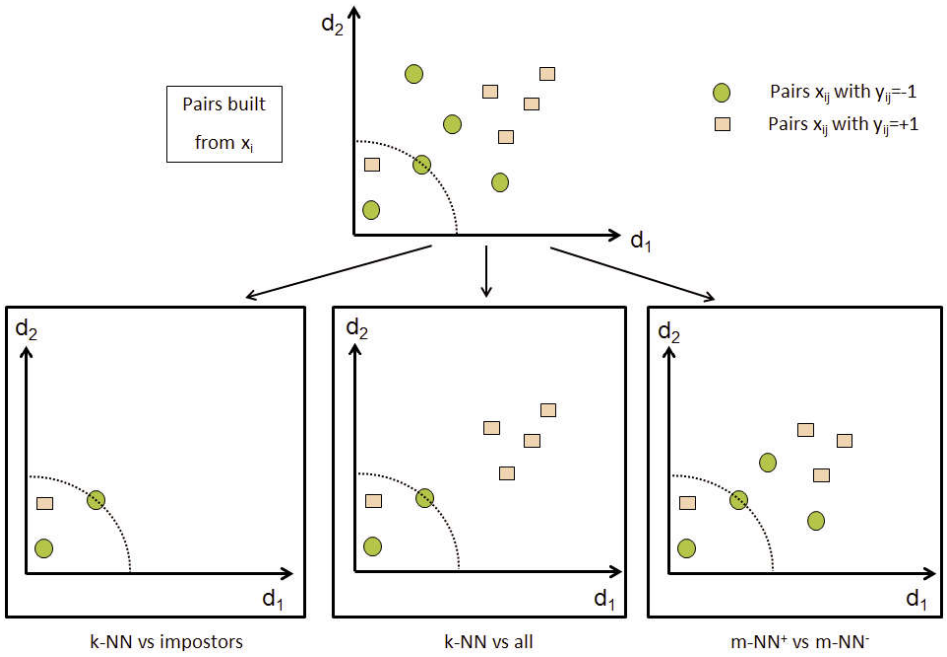

Figure 3.5: Example of a k-NN problem with $k = 2$. From the embedding of time series $\mathbf{x}_i$ in the dissimilarity space (top), three different strategies (bottom) for pairwise training set X_p definition: k-NN vs impostor (left), k-NN vs all (middle) and m-NN$^+$ vs m-NN$^-$ (right) with $m = 4$.

during the testing phase. Note that computationally, this strategy is more expensive than the k-**NN vs impostors** strategy but less expensive than k-**NN vs all** strategy.

3.3.3 Relationship with LMNN and M^{2}TML

Even if the optimization problem in Eq. 3.4 is similar to the one of Large Margin Nearest Neighbors (LMNN) proposed by Weinberger & Saul in [WS09a] (Eq. 2.24), let recall the main points:

- The sets $Pull_i$ and $Push_i$ are defined according the k-**NN vs impostors** strategy.
- The sets $Pull_i$ and $Push_i$ can be unbalanced.
- The initial distance is the Euclidean distance.
- The sets $Pull_i$ and $Push_i$ are defined and fixed during the optimization process according to the considered initial metric.
- The learnt metric D is a Mahalanobis distance.

Our proposition (M^{2}TML) differs from LMNN and we detailed here the main points:

- The sets $Pull_i$ and $Push_i$ are defined according the m-**NN**$^+$ **vs** m-**NN**$^-$ strategy with $m = \alpha.k$, $\alpha \geq 1$.
- The sets $Pull_i$ and $Push_i$ are balanced.
- The initial distance is the Euclidean distance.
- The sets $Pull_i$ and $Push_i$ are defined and fixed during the optimization process according to the considered initial metric. However, the m-neighborhood is larger than the k-neighborhood allowing a better generalization of the learnt metric D.
- The learnt metric D is a combined metric of several unimodal metrics measured at different scales.

In the following, we propose different regularizers for the pull term $R(Pull_i)$. 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 which aims first to learn an hyperplane that separates the sets $Pull_i$ and $Push_i$, and then we propose a solution to define the metric D from this hyperplane. Finally, we sum up the retained solution (SVM-based solution) and gives the main steps of the algorithm.

3.4 Linear formalization of the combined metric

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 and use the pairwise notation for simplification purpose:

$$D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_{ij}) = \mathbf{w}^T \mathbf{x}_{ij} = \sum_{h=1}^{p} w_h.d_h(\mathbf{x}_i, \mathbf{x}_j) \tag{3.13}$$

where $\mathbf{w} = [w_1, \ldots, w_p]^T$ is the vector of weights w_h. Optimizing the metric D is equivalent

to optimizing the weight vector $\mathbf{w}$. Eq. 3.4 leads to the following optimization problem:

$$\begin{aligned} \underset{\mathbf{w},\xi}{\operatorname{argmin}} & \left\{ \underbrace{\sum_{\substack{i \\ j \in Pull_i}} \mathbf{w}^T \mathbf{x}_{ij}}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1+y_{il}}{2} \xi_{ijl}}_{push} \right\} \\ \text{s.t. } & \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ & \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \end{aligned} \tag{3.14}$$

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, and $Pull_i$ and $Push_i$ are defined in Eq. 3.11. We recall that the sets $Pull_i$ and $Push_i$ are defined using the m-**NN**$^+$ **vs** m-**NN**$^-$ strategy. 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 $\mathbf{x}_{ij}$.

Concerning the properties of the learnt metric D, to ensure positivity, as the metrics $d_1, \ldots, d_p$ are dissimilarity measures ($d_h \geq 0$), it requires that:

$$\forall h = 1...p, \qquad w_h \geq 0 \tag{3.15}$$

This constraint is added up to the optimization problem in Eq. 3.14. As the metric D is defined as a linear combination of dissimilarity measures $d_1, \ldots, d_p$, the symmetry and distinguishability properties are ensured:

$$d_h(\mathbf{x}_i, \mathbf{x}_j) = d_h(\mathbf{x}_j, \mathbf{x}_i) \Leftrightarrow D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i) \tag{3.16}$$

$$d_h(\mathbf{x}_i, \mathbf{x}_i) = 0 \Rightarrow D(\mathbf{x}_i, \mathbf{x}_i) = 0 \tag{3.17}$$

Note that reciprocity for distinguishability ($D(\mathbf{x}_i, \mathbf{x}_i) = 0 \Rightarrow d_h(\mathbf{x}_i, \mathbf{x}_i) = 0$) is ensured if the w_h are not all equal to zero. The triangular inequality property of D has not been studied and is not thus ensured.

3.5 Quadratic formalization of the combined metric

In this section, we define the problem of learning a combined metric 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 Dual formalization

The formulation in Eq. 3.14 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 regularizer $R(Pull)$ in the objective function of Eq. 3.14. Two solutions are studied:

$$1. \quad R(Pull_i) = \frac{1}{2}\sum_{h=1}^{p} \sum_{\substack{i \\ j \in Pull_i}} \left(w_h d_h(\mathbf{x}_{ij})\right)^2 \tag{3.18}$$

$$2. \quad R(Pull_i) = \frac{1}{2}\sum_{h=1}^{p} \left(\sum_{\substack{i \\ j \in Pull_i}} w_h d_h(\mathbf{x}_{ij}) \right)^2 = \frac{1}{2} m.n \sum_{h=1}^{p} \left(w_h \bar{d_h}\right)^2 \tag{3.19}$$

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

$$R(Pull_i) = \frac{1}{2}\mathbf{w}^T \mathbf{M} \mathbf{w} \tag{3.20}$$

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

$$1. \quad \mathbf{M} = Diag(\mathbf{X}_{pull}^T \mathbf{X}_{pull}) = \begin{bmatrix} \sum\limits_{\substack{i \\ j \in Pull_i}} d_1^2(\mathbf{x}_{ij}) & & 0 \\ & \ddots & \\ 0 & & \sum\limits_{\substack{i \\ j \in Pull_i}} d_p^2(\mathbf{x}_{ij}) \end{bmatrix} \tag{3.21}$$

$$2. \quad \mathbf{M} = Diag(\bar{\mathbf{X}}) Diag(\bar{\mathbf{X}}) = \begin{bmatrix} \bar{d_1}^2 & & 0 \\ & \ddots & \\ 0 & & \bar{d_p}^2 \end{bmatrix} \tag{3.22}$$

where $\mathbf{X}_{pull}$ be a $(m.n) \times p$ matrix containing the vector $\mathbf{x}_{ij} \in Pull_i$ and $\bar{\mathbf{X}}$ is a $p \times 1$ matrix containing the mean of the metrics $\bar{d_h}$.

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

$$\begin{aligned} \underset{\mathbf{w},\xi}{\operatorname{argmin}} & \left\{ \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w}}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1+y_{il}}{2}\xi_{ijl}}_{push} \right\} \\ \text{s.t. } & \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ & \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \end{aligned} \tag{3.23}$$

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 $\mathbf{r}$:

$$\begin{aligned} L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) = & \frac{1}{2}\mathbf{w}^T\mathbf{M}\mathbf{w} + C\sum_{ijl}\frac{1+y_{il}}{2}\xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\ & - \sum_{ijl}\alpha_{ijl}\left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl})\right) \end{aligned} \tag{3.24}$$

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. 3.21 & 3.22), it is thus inversible. The equations leads to:

$$\mathbf{w} = \mathbf{M}^{-1}\sum_{ijl}\alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.25}$$

$$r_{ijl} = C - \alpha_{ijl} \tag{3.26}$$

Substituting Eq. 3.25 and 3.26 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.24, we get the dual formulation[1]:

[1] complete details of the calculations in Appendix D

$$\begin{aligned} &\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1} (\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \\ &\text{s.t. } \forall i = 1, \ldots, n, \forall j \in Pull_i, l \in Push_i, \\ &0 \leq \alpha_{ijl} \leq C \end{aligned} \tag{3.27}$$

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

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1}}_{\mathbf{w}^T} \mathbf{x}_{i'j'} \tag{3.28}$$

$$D(\mathbf{x}_{i'j'}) = \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{il}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity to } Push_i} - \underbrace{\sum_{ijl} \alpha_{ijl} \mathbf{x}_{ij}^T \mathbf{M}^{-1} \mathbf{x}_{i'j'}}_{\text{similarity to } Pull_i} \tag{3.29}$$

Similarly to SVM, 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 (Eq. 3.29): 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: the first one computes the similarity of a new pairwise vector $\mathbf{x}_{i'j'}$ to the vectors $\mathbf{x}_{il}$ of the push set $Push_i$; the second term computes the similarity of a new pairwise vector $\mathbf{x}_{i'j'}$ to the vectors $\mathbf{x}_{ij}$ of the pull set $Pull_i$. The direction of the weight vector $\mathbf{w}$ is determined by the mean direction of the sets $Push_i$ and $Pull_i$.

Concerning the properties of the metric D, as in the linear formalization, to ensure the positivity properties, it requires that $\forall h = 1, \ldots, p, w_h \geq 0$. The constraints cannot been added to the optimization problem because the dual formulation wouldn't depends on only inner products. As the metric D relies on the difference of two similarity terms, the positivity is not always ensured, *e.g.*, the similarity of the pull term is greater than the similarity of the push term. As d_h are dissimilarity measure ($d_h(\mathbf{x}_{i'j'}) = d_h(\mathbf{x}_{j'i'}) \Leftrightarrow \mathbf{x}_{i'j'} = \mathbf{x}_{j'i'}$), symmetry property for D is ensured:

$$\begin{aligned} &D(\mathbf{x}_{j'i'}) = \sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1} \mathbf{x}_{j'i'} = \sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T \mathbf{M}^{-1} \mathbf{x}_{i'j'} \\ &D(\mathbf{x}_{j'i'}) = D(\mathbf{x}_{i'j'}) \end{aligned}$$

Similarly to the linear formalization, for distinguishability, the property is ensured for $\mathbf{x}_{i'j'} = 0 \Rightarrow D(\mathbf{x}_{j'i'}) = 0$. However, the reciprocity is not ensured as D is expressed as a difference of two terms, there exists other possibilities for $D(\mathbf{x}_{j'i'}) = 0$, *e.g.*, if $\forall i, j \in Pull_i, l \in Push_i, \mathbf{x}_{il} = \mathbf{x}_{ij}$

3.5.2 Extension to non-linear function of D

The above formula can extended to non-linear function for the metric D. The dual formulation in Eq. 3.27 only relies on the inner product $(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'})^T \mathbf{M}^{-1} (\mathbf{x}_{il} - \mathbf{x}_{ij})$. We can hence apply the kernel trick to find non-linear solutions for D. 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 $\mathbf{M}^{-\frac{1}{2}}$:

$$1. \quad \mathbf{M}^{-\frac{1}{2}} = Diag(\mathbf{X}_{pull}^T \mathbf{X}_{pull}) = \begin{bmatrix} \frac{1}{\sum\limits_{\substack{i \\ j \in Pull_i}} d_1^2(\mathbf{x}_{ij})} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\sum\limits_{\substack{i \\ j \in Pull_i}} d_p^2(\mathbf{x}_{ij})} \end{bmatrix} \tag{3.30}$$

$$2. \quad \mathbf{M}^{-\frac{1}{2}} = Diag(\bar{\mathbf{X}}) Diag(\bar{\mathbf{X}}) = \begin{bmatrix} \frac{1}{\bar{d_1}^2} & & 0 \\ & \ddots & \\ 0 & & \frac{1}{\bar{d_p}^2} \end{bmatrix} \tag{3.31}$$

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.

From this, the inner product in Eq. 3.27 can be replaced by a kernel:

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

By replacing the inner product by a kernel back into Eq. 3.28, we obtain:

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl} \kappa(\mathbf{M}^{-\frac{1}{2}} (\mathbf{x}_{il} - \mathbf{x}_{ij}); \mathbf{M}^{-\frac{1}{2}} \mathbf{x}_{i'j'}) \tag{3.32}$$

Note that as $\phi(\mathbf{0})$ doesn't meet the origin in the feature space $\mathscr{H}$, the distance in Eq. 3.32 needs to be centered with respect to $\phi(\mathbf{0})$:

$$\begin{aligned} D(\mathbf{x}_{i'j'}) = &\sum_{ijl} \alpha_{ijl} \kappa \left(\mathbf{M}^{-\frac{1}{2}} (\mathbf{x}_{il} - \mathbf{x}_{ij}); \mathbf{M}^{-\frac{1}{2}} (\mathbf{x}_{i'j'} - \mathbf{x}_{ij}) \right) \\ &- \sum_{ijl} \alpha_{ijl} \kappa \left(\mathbf{M}^{-\frac{1}{2}} (\mathbf{x}_{il} - \mathbf{x}_{ij}); \mathbf{M}^{-\frac{1}{2}} (\mathbf{0} - \mathbf{x}_{ij}) \right) \end{aligned} \tag{3.33}$$

$$D(\mathbf{x}_{i'j'}) = \overbrace{\sum_{ijl} \alpha_{ijl} K(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Push_i} - \overbrace{\sum_{ijl} \alpha_{ijl} K(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{i'j'})}^{\text{similarity of } \mathbf{x}_{i'j'} \text{ to } Pull_i}$$
$$\underbrace{- \sum_{ijl} \alpha_{ijl} K(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{il}; \mathbf{M}^{-\frac{1}{2}}\mathbf{0})}_{\text{similarity of } \mathbf{0} \text{ to } Push_i} + \underbrace{\sum_{ijl} \alpha_{ijl} K(\mathbf{M}^{-\frac{1}{2}}\mathbf{x}_{ij}; \mathbf{M}^{-\frac{1}{2}}\mathbf{0})}_{\text{similarity of } \mathbf{0} \text{ to } Pull_i} \tag{3.34}$$

Similarly to the linear case, the metric D can be interpreted as a sum and difference of 4 similarity terms: the first one is the similarity of the new vector $\mathbf{x}_{i'j'}$ to the pairs of $Push_i$; the second is the similarity of the new vector $\mathbf{x}_{i'j'}$ to the pairs of $Pull_i$; the third one is the similarity of the null vector $\mathbf{0}$ to the pairs of $Push_i$; the second is the similarity of the null vector $\mathbf{0}$ to the pairs of $Pull_i$.

Concerning the properties of the metric D, as a sum and difference of 4 terms, positivity is not ensured. Symmetry property is ensured as $\mathbf{x}_{i'j'} = \mathbf{x}_{j'i'}$. Distinguishability is ensured for $\mathbf{x}_{i'j'} = 0 \Rightarrow D(\mathbf{x}_{i'j'}) = 0$ but the reciprocity is not ensured as a sum and difference of 4 terms

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. 3.35 for $\mathbf{w}$ and b solves a similar problem with a quadratic regularization in Eq. 3.23 for $D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(\mathbf{w}^T\mathbf{x}_{ij} + b)$.

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i \text{ or} \\ j \in Push_i}} p_i \xi_{ij} \right\} \\ &\text{s.t. } \forall i, j \in Pull_i \text{ or } j \in Push_i : \\ &y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \end{aligned} \tag{3.35}$$

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 $Push_i$:

$$\begin{aligned}
&\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{\substack{i \\ j \in Pull_i}} p_i^- \xi_{ij} + C \sum_{\substack{i \\ l \in Push_i}} p_i^+ \xi_{il} \right\} \\
&\text{s.t.} \ : \\
&\forall i, j \in Pull_i : y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\
&\forall i, l \in Push_i : y_{il}(\mathbf{w}^T \mathbf{x}_{il} + b) \geq 1 - \xi_{il}
\end{aligned} \tag{3.36}$$

where p_i^- and p_i^+ are the weight factors for push 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 $Push_i$:

$$p_i^+ = \frac{k}{2} = \sum_{j \in Pull_i} \frac{1}{2} \tag{3.37}$$

$$p_i^- = \frac{L}{2} = \frac{1}{2} \sum_{l} \frac{1 + y_{il}}{2} \tag{3.38}$$

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

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

$$\forall i, j \in Pull_i, l \in Push_i : D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijl}$$

- 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:

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

- Both problems share the same loss/push term: $\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1+y_{il}}{2} \xi_{ijl}$

- 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. 3.18 & 3.19)

- 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 a hyperplane, the constraint $w_h \geq 0$ does not hold. Symmetry is ensured as $\mathbf{x}_{ij} = \mathbf{x}_{ji}$. Distinguishability is ensured if the w_h are not all equal to zero.

3.6 SVM-based formalization of the combined metric

In this section, we present a solution based on SVM where the form of the metric D is not known *a priori*. First, we learn an hyperplane that separates the sets $Pull_i$ and $Push_i$. Secondly, we propose a solution to build a metric D from the obtained hyperplane so that the metric D satisfy the required properties of a metric. 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 an SVM framework as follows in the dissimilarity space $\mathscr{E}$:

$$\begin{aligned} &\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} \frac{1}{2}||\mathbf{w}||_2^2 + C\sum_{i,j}\xi_{ij} \\ &\textbf{s.t. } y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\xi_{ij} \geq 0 \end{aligned} \tag{3.39}$$

In the linear case, a L_1 regularization in Eq. 3.39 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:

$$\begin{aligned} &\underset{\mathbf{w},b,\boldsymbol{\xi}}{\operatorname{argmin}} ||\mathbf{w}||_1 + C\sum_{i,j}\xi_{ij} \\ &\textbf{s.t. } y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \\ &\xi_{ij} \geq 0 \end{aligned} \tag{3.40}$$

Note that in practice, the local neighborhoods for each sample $\mathbf{x}_i$ ($Pull_i + Push_i$) 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 Definition of the learnt metric in the linear case

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 propositions to define the learned metric D: Scalar product, Projected norm, Exponential transformation.

Scalar product and Projected norm

First, the learned metric D can be defined as the distance to the hyperplane:

$$D(\mathbf{x}_{i,test}) = \mathbf{w}^T \mathbf{x}_{i,test} + b \tag{3.41}$$

The obtained metric D doesn't necessarily satisfy the distinguishability ($D(\mathbf{x}_{ii} = 0)$) and positivity ($D(\mathbf{x}_{ij} \geq 0)$) property, especially when push pairs $Push_i$ are situated nearer to the origin point than pull pairs $Pull_i$ (Fig. 3.6).

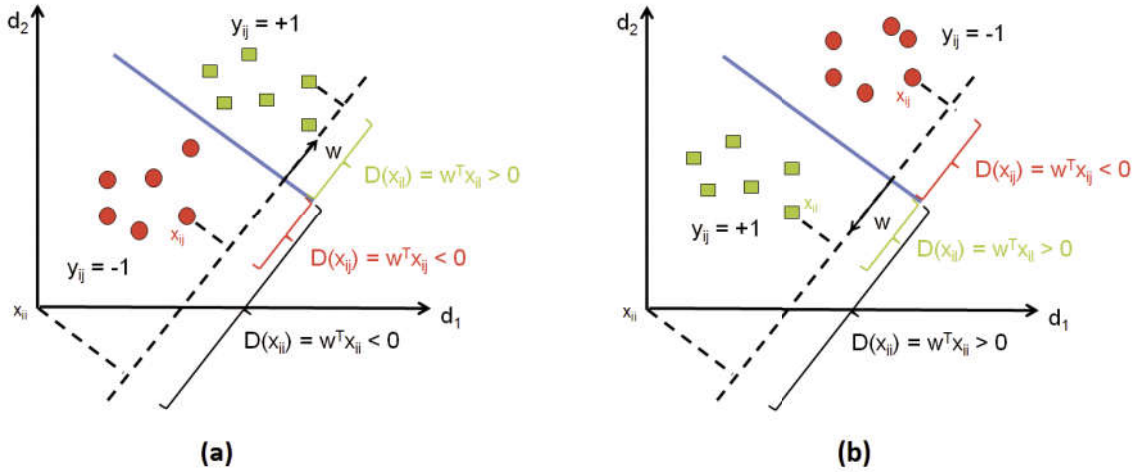

Figure 3.6: Example of SVM solutions and of the resulting metric D defined by a scalar product. Fig. (a) represents common expected configuration where pull pairs $Pull_i$ are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b), the vector $\mathbf{w} = [-1 - 1]^T$ indicates that $Push_i$ pairs are on the side of the origin point. For the two configurations, two problems arises: First, for pull pairs $Pull_i$, $D(\mathbf{x}_{ij}) \leq 0$. Secondly, for the origin point $\mathbf{x}_{ii}$, we obtain $D(\mathbf{x}_{ii}) \neq 0$.

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

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})|| = ||\mathbf{w}^T \mathbf{x}_{i,test}|| \tag{3.42}$$

Although the norm $||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})||$ 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.

Exponential transformation

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:

$$D(\mathbf{x}_{i,test}) = ||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})||. \exp(\lambda[\mathbf{w}\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test}) + b]_+) \quad \lambda > 0 \tag{3.43}$$

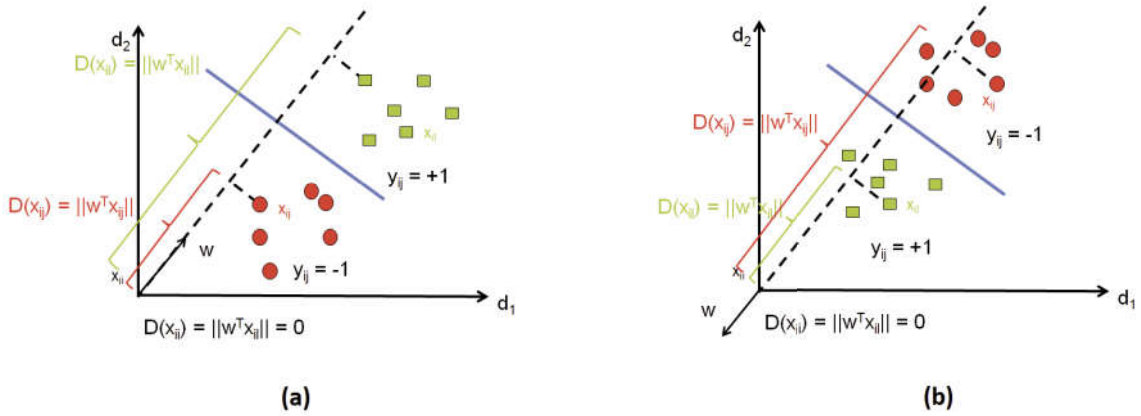

Figure 3.7: Example of SVM solutions and of the resulting metric D defined by the norm of the projection on $\mathbf{w}$. Fig. (a) represents common expected configuration where pull pairs $Pull_i$ are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b), the vector $\mathbf{w}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

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

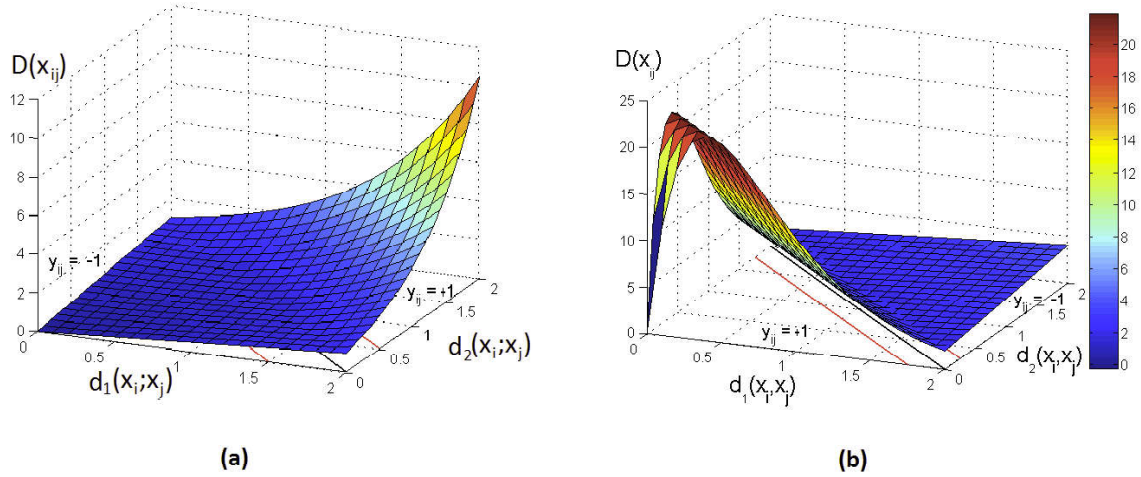

Figure 3.8: The behavior of the learned metric D ($p = 2$; $\lambda = 2.5$) with respect to common (a) and challenging (b) configurations of pull and push pairs.

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 pull pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin. The

dissimilarity increases proportionally to the norm in the 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 push pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of different classes) are situated in the same side as the origin. That means that time series $\mathbf{x}_j$ that are of different classes from $\mathbf{x}_i$ are closer to $\mathbf{x}_i$ than its nearest neighbors. They are thus impostors. The dissimilarity behaves proportionally to the norm on the 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 push pairs, *i.e.*, pairs of norm zero labeled differently.

3.6.3 Definition of the learnt metric in the non-linear case

The above solution holds true for any kernel K and allows to extend the dissimilarity D given in Eq. 3.43 to non linearly separable pull and push pairs. Let K be a kernel defined in the pairwise space $\mathscr{E}$ and the related Hilbert space (feature space) $\mathscr{H}$. For a non linear combination function of the metrics $d_h, h = 1, \ldots, p$ in $\mathscr{E}$, we define the dissimilarity measure $D_{\mathscr{H}}$ in the feature space $\mathscr{H}$ as:

$$\begin{aligned} D_{\mathscr{H}}(\mathbf{x}_{i,test}) =& (||\mathbf{P}_{\mathbf{w}}(\phi(\mathbf{x}_{i,test}))|| - ||\mathbf{P}_{\mathbf{w}}(\phi(\mathbf{0}))||). \\ & \exp\left(\lambda \left[\sum_{ij} y_{ij}\alpha_{ij}K(\mathbf{x}_{ij}, \mathbf{x}_{i,test}) + b\right]_+\right) \qquad \lambda > 0 \end{aligned} \tag{3.44}$$

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

$$||\mathbf{P}_{\mathbf{w}}(\mathbf{x}_{i,test})|| = \frac{\sum_{ij} y_{ij}\alpha_{ij}K(\mathbf{x}_{ij}, \mathbf{x}_{i,test})}{\sqrt{\sum_{ijkl} \alpha_{ij}\alpha_{kl}y_{ij}y_{kl}K(\mathbf{x}_{ij}, \mathbf{x}_{kl})}} \tag{3.45}$$

Note that as $\phi(\mathbf{0})$ doesn't meet the origin in the feature space $\mathscr{H}$, the norms in Eq. 3.44 are centered with respect to $\phi(\mathbf{0})$.

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

3.7 SVM-based solution and algorithm for metric learning

3.7.1 Algorithm

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

Note sur le neighborhood scaling en test?

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

1: Input: $X = \{\mathbf{x}_i, y_i\}_{i=1}^N$ N labeled time series
$d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 3.2
a kernel K
2: Output: the learned dissimilarity D or $D_{\mathscr{H}}$ depending of K
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 positive m-NN$^+$ and negative m-NN$^-$ pairs and scale the radii to 1.
5: Train a SVM for a large margin classifier between m-NN$^+$ and m-NN$^-$ (Eq. 3.39)
6: *Dissimilarity definition*
Consider Eq. 3.43 (resp. Eq. 3.44) 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: $X = \{\mathbf{x}_i, y_i\}_{i=1}^N$ N labeled time series
$\{\mathbf{x}_{test}, y_{test}\}$ a labeled time series to test
$d_1, ..., d_p$ metrics as described in Eqs. 2.1, 2.4, 2.6, 3.2
the learned dissimilarity D or $D_{\mathscr{H}}$ depending of the kernel K
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. 3.43 (resp. Eq. 3.44) 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 $\mathbf{x}_{test}$

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

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

$$r_i = \max_{\mathbf{x}_{ij} \in Pull_i} ||\mathbf{x}_{ij}||_2 \tag{3.46}$$

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:

$$\mathbf{x}_{ij}^{norm} = \left[\frac{d_1(\mathbf{x}_{ij})}{r_i}, \ldots, \frac{d_p(\mathbf{x}_{ij})}{r_i} \right]^T \tag{3.47}$$

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.

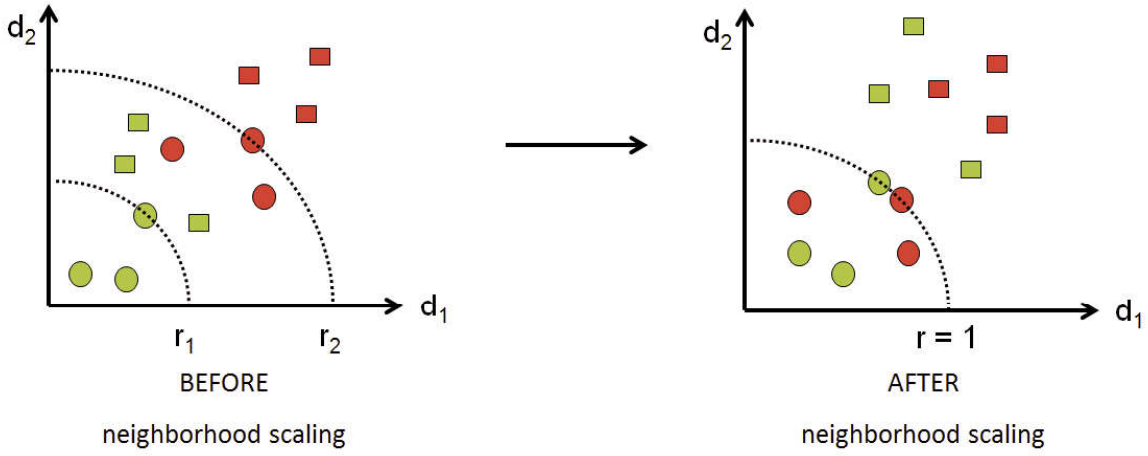

Figure 3.9: Effect of neighborhood scaling before (left) and after (right) on the neighborhood of two time series $\mathbf{x}_1$ (green) and $\mathbf{x}_2$ (red). Circle represent pull pairs $Pull_i$ and square represents push pairs $Push_i$ for $m = 2$ neighbors. Before scaling, the problem is not linearly separable. The spread of each neighborhood are not comparable. After scaling, the target neighborhood becomes comparable and in this example, the problem becomes linearly separable between the circles and the squares.

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

3.7.2 Extension to regression problems

In the dissimilarity space, each vector $\mathbf{x}_{ij}$ can be labeled y_{ij} by following the rule: "if $\mathbf{x}_i$ and $\mathbf{x}_j$ are similar ($Pull_i$), the vector $\mathbf{x}_{ij}$ is labeled -1; and $+1$ otherwise ($Push_i$)."
Until here, we solve the metric learning for classification problems. The concept of similarity between samples $\mathbf{x}_i$ and $\mathbf{x}_j$ is driven by the class label y_i and y_j in the original space:

$$y_{ij} = \begin{cases} -1 & \text{if } y_i = y_j \\ +1 & \text{if } y_i \neq y_j \end{cases} \tag{3.48}$$

For regression problems, each sample $\mathbf{x}_i$ is assigned to a continuous value y_i. Two approaches are possible to define the similarity concept. The first one discretizes the continuous space of values of the labels y_i to create classes. One possible discretization bins the label y_i into Q intervals. Each interval becomes a class which associated value can be set for example as the mean or median value of the interval. Then, the classification framework is used to define the pairwise label y_{ij}. This approach may leads to border effects between the classes. For instance, two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ that are close to a frontier and that are on different sides of the border will be considered as different. Moreover, a new sample $\mathbf{x}_j$ will have its labels y_j assigned to a class and not a real continuous value.
The second approach considers the continuous value of y_i, computes the absolute difference $|y_i - y_j|$ between the labels y_i and y_j, and compare this value to a threshold ϵ. Geometrically, a tube of size ϵ around each value of y_i is built. Two samples $\mathbf{x}_i$ and $\mathbf{x}_j$ are considered as similar if the absolute difference $|y_i - y_j|$ is lower than ϵ (Fig. 3.10):

$$y_{ij} = \begin{cases} -1 & \text{if } |y_i - y_j| \leq \epsilon \\ +1 & \text{otherwise} \end{cases} \tag{3.49}$$

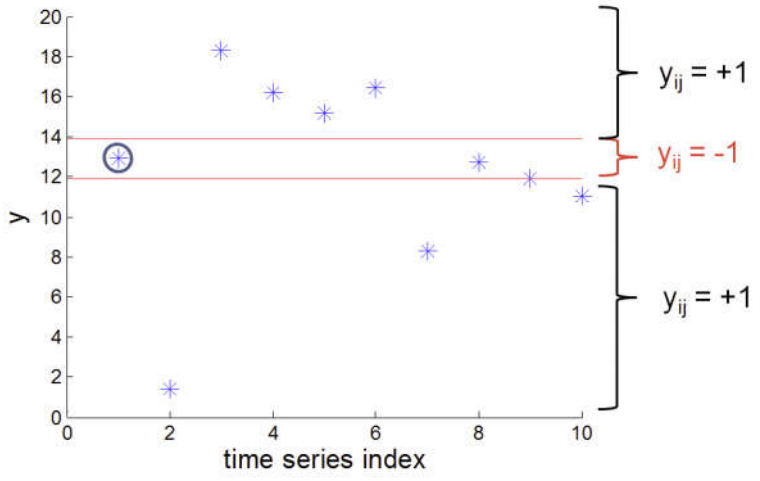

Figure 3.10: Example of pairwise label definition using an ϵ-tube (red lines) around the time series $\mathbf{x}_i$ (circled in blue). For, time series $\mathbf{x}_j$ that falls into the tube, the pairwise label is $y_{ij} = -1$ (similar) and outside of the tube, $y_{ij} = +1$ (not similar).

3.8 Geometric interpretation

A changer avec les retours de Sylvain. Michèle pense que l'état, cette section est dure à comprendre. D'après Michèle, il faut 1) soit prendre + de place pour expliquer la signification géométrique 2) ou soit ne pas mettre cette partie car étant compliquée, cela pourrait nuire au lecteur. Qu'en penses tu Ahlame?

In this section, we give a geometric understanding of the differences between LP/QP resolution (left) and SVM-based resolution (right). Fig. 3.12 shows the Linear Programming (LP) and SVM resolutions of a k-NN problem with $k = 2$ neighborhoods.

For LP, the problem is solved for each neighborhood (blue and red) independently as shown in Fig. 3.11. We recall that LP/QP resolutions, support vectors are triplets of time series made of a target pair $\mathbf{x}_{ij}$ and a pair of different classes $\mathbf{x}_{il}$ (black arrows). Support vectors represent triplet which resulting distance $D(\mathbf{x}_{ij}, \mathbf{x}_{il})$ are the lowest. The optimization problem tends to maximize the margin between these triplets. The global solution (Fig. 3.12 (left)) is a compromise of all of the considered margins. In this case, the global margin is equal to one of the local margin. Note that the global LP solution is not always the same as the best local solution. For SVM-based resolution (Fig. 3.12 (right)), the problem involves all pairs and the margin is optimized so that pairs $\mathbf{x}_{ij}$ and $\mathbf{x}_{il}$ are globally separated.

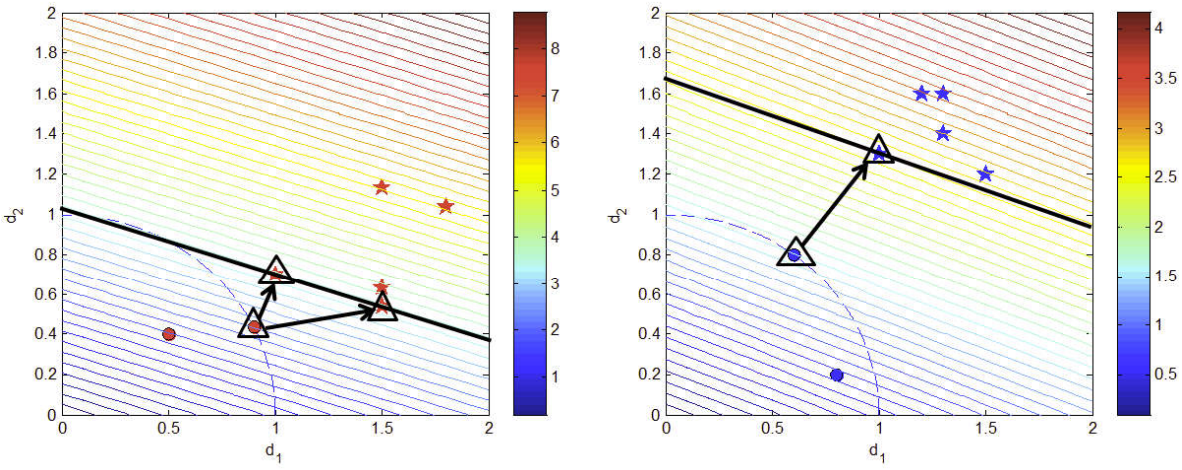

Figure 3.11: Solutions found by solving the LP problem for $k = 2$ neighborhood. Positive pairs (different classes) are indicated in stars and negative pairs (target pairs) are indicated in circle. Red and blue lines shows the margin when solving the problem for each neighborhood (red and blue points) separately. Support vector are indicated in black triangles: in the red neighborhood (left), 2 support vectors are retained and in the blue neighborhood (right), only one support vector is necessary.

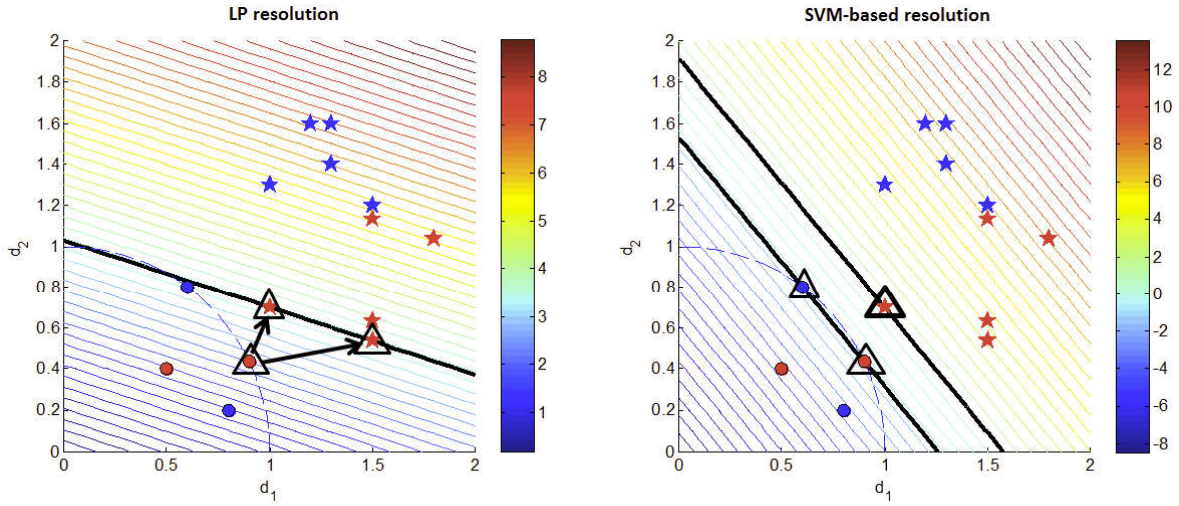

Figure 3.12: Solutions found by solving the LP problem (left) and the SVM problem (right). The global margin is indicated in black and the metric is represented in color levels. Support vectors made of triplets are indicated in black triangles. For the SVM, the black lines indicates the SVM canonical hyperplane where the support vector lies (black triangles).

3.9 Conclusion of the chapter

To learn a combined metric, we propose in this chapter to embed time series into a dissimilarity space whose dimensions are the different unimodal and uniscale metric. The multi-modal and multi-scale metric learning problem can be formalized as a problem of learning a function in the dissimilarity space. Among all the possible function, we aim at a subset of function that ensures the required properties for a metric. 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 during the testing phase.

From the general formalization, we propose three different formalizations (Linear, Quadratic, SVM-based) in which the regularization term and the definition of the metric is known *a priori* or not. Table 3.1 sums up the main pros and cons of each formalization.

	Linear formalization	Quadratic formalization	SVM-based formalization
Linear	Yes	Yes	Yes
Non-linear extension	No	Yes	Yes
Exact/Approximation resolution	Exact	Exact	Approximation
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
Distinguishability of D	Yes (under conditions)	Yes (under conditions)	Yes (always)
Triangular inequality of D	Not studied	Not studied	Not studied

Table 3.1: The different formalizations for Metric Learning in Dissimilarity space

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 the objective of the SVM is to separate negative from positive classes.

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