Chapter 3

Metric Learning in dissimilarity space: formalization

Sommaire

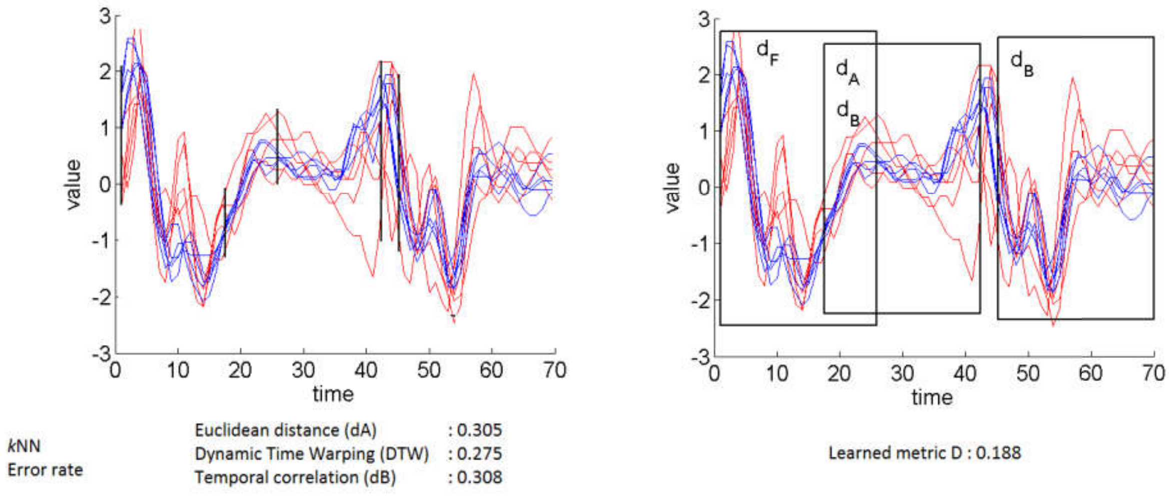
In this chapter, we formalize the problem of Metric Learning in Dissimilarity space (MLD). We first motivate the problem of learning a metric that combines several metrics at different scales for a robust k-NN classifier. Secondly, we introduce the concept of dissimilarity space. Finally, we formalize the problem of learning a combined metric in the initial space as a standard metric learning problem into the dissimilarity space and propose three possible formulations: Linear programming, Quadratic programming and SVM-based approximation.

3.1 Motivations

The definition of a metric to compare samples is a fundamental issue in data analysis, pattern recognition 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

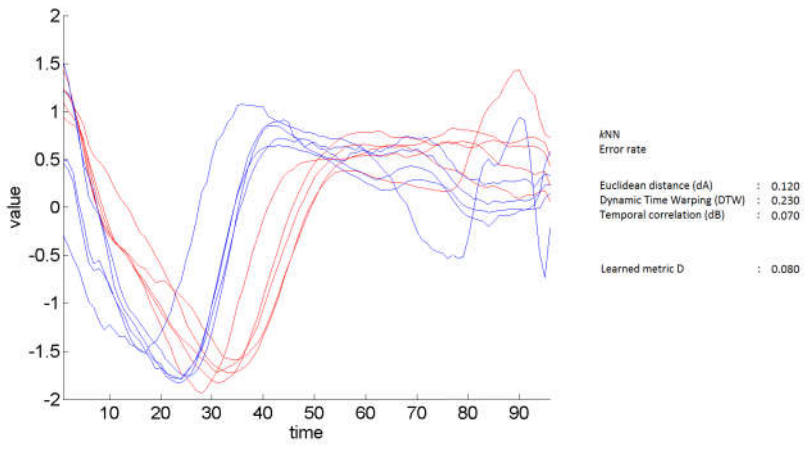
based on one temporal modality. For amplitude-based comparison, measures cover variants of Mahalanobis distance or the dynamic time warping (DTW) to cope with delays [BC94b]; [Rab89]; [SC78b]; [KL83]. Other propositions refer to temporal correlations or derivative dynamic time warping for behavior-based 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.
In some applications, the most discriminative characteristic between time series of different classes can be localized on a smaller part of the signal. Involving totally or partially time series elements, rather than systematically the whole elements should be taken into consideration in the metric definition, a crucial key to localize discriminative features. In other applications, the combinations of several factors (modality, scale) can improve the performance of the classifier. Thus, there is a need for combined metrics.
Some works propose to combine several modalities through a priori models as in [DCDG10]; [DCA12]; [SB08]. Ideally, a combined metric should answer two scenarios depending on the datasets: 1) combines several modalities at several scales; 2) select one modality at one particular scale and thus, coming back to a uni-modal and uni-scale metric framework. Figs. 3.1 and 3.2 shows these two cases on two dataset examples used in classification of univariate time series. For SonyAIBO dataset (Fig. 3.1), by learning the metric, several modalities (frequential d_F, behavior d_B and amplitude d_A) at different locally temporal intervals are combined together. Thanks to this combination, the error of the 1-NN classifier is decreased: global uni-modal metrics achieves 0.305 for d_A, 0.308 for d_B, 0.258 for d_F; and the combined metric achieves a score of 0.188. For ECG200 dataset (Fig. 3.2), the learned metric mainly includes the global behavior component d_B and the error rate remains statistically the same. More detailed explanations will be given in Chapter 5.

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, and that satisfies metric properties (Section 2.2).
Metric learning can be defined as learning, from the data and for a task, a pairwise function (i.e. a similarity, dissimilarity or a distance) to make closer samples that are expected to be similar, and far away those expected to be dissimilar. Similar and dissimilar samples, are inherently task- and application-dependent, generally given a priori and fixed during the learning process. Metric learning has become an active area of research in last decades for various machine learning problems (supervised, semi-supervised, unsupervised, online learning) and has received many interests in its theoretical background (generalization guarantees) [BHS13]. From the surge of recent research in metric learning, one can identify mainly two categories: the linear and non linear approaches. The former is the most popular, it defines the majority of the propositions, and focuses mainly on the Mahalanobis distance learning [WS09a]. The latter addresses non linear metric learning which aims to capture non linear structure in the data. In Kernel Principal Component Analysis (KPCA) [ZY10]; [Cha+10], the aim is to project the data into a non linear feature space and learn the metric in that projected space.



*k*NN Error rate	Euclidean distance (dA)	: 0.305
	Dynamic Time Warping (DTW)	: 0.275
	Temporal correlation (dB)	: 0.308
	Learned metric D	: 0.188

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



*k*NN Error rate		
	Euclidean distance (dA)	: 0.120
	Dynamic Time Warping (DTW)	: 0.230
	Temporal correlation (dB)	: 0.070
	Learned metric D	: 0.080

Figure 3.2: ECG200 dataset and error rate using a kNN with $k = 1$ with standard metrics (Euclidean distance, Dynamic Time Warping, temporal correlation) and a learned combined metric. For the learned combined metric D, the major discriminant feature is the behavior-based metric d_B (90% of D) computed on a global scale (including all time series elements).

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. Bellet & al. enlightened in [BHS13] perspectives for metric learning, especially, the learning of richer metrics that could take into account of the multi-modality within the data.

We propose in this work to learn a multi-modal and multi-scale temporal metric for a robust k-NN classifier. For this, the main idea is to embed time series into a dissimilarity space [PPD02]; [DP12] where a linear function combining several modalities at different temporal scales can be learned, driven jointly by a SVM and nearest neighbors metric learning framework [WS09b]. Thanks to the "kernel trick", the proposed solution is extended to non-linear temporal metric learning context. A sparse and interpretable variant of the algorithm 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 Metric Learning problem in the Dissimilarity space (MDL) and formalize the problem under three formulations: Linear Programming, Quadratic Programming, SVM approximation. Note that these formulations doesn't concern only time series and can be applied to any type of data. In the next chapter, we detailed our proposed solution to learn a multi-modal and multi-scale temporal metric (M^{2}TML) for a robust k-NN classifier.

3.2 Dissimilarity space

Let $d_1, ..., d_h ..., d_p$ be p given metrics that allow to compare samples. 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 that combines the p metrics in order to optimize the performance of a k-NN classifier. Formally:

$$D = f(d_1, \ldots, d_p) \tag{3.1}$$

In this section, we first introduce the dissimilarity space. Then, we give some interpretations in the dissimilarity space.

3.2.1 Pairwise embedding

The computation of a metric d, and of course 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**. In this new space, illustrated in Fig. 3.3, a vector $\mathbf{x}_{ij}$ represents a pair of time series $(\mathbf{x}_i, \mathbf{x}_j)$ described by the p unimodal metrics d_h: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$. We denote N the number of pairwise vectors $\mathbf{x}_{ij}$ generated by this embedding.

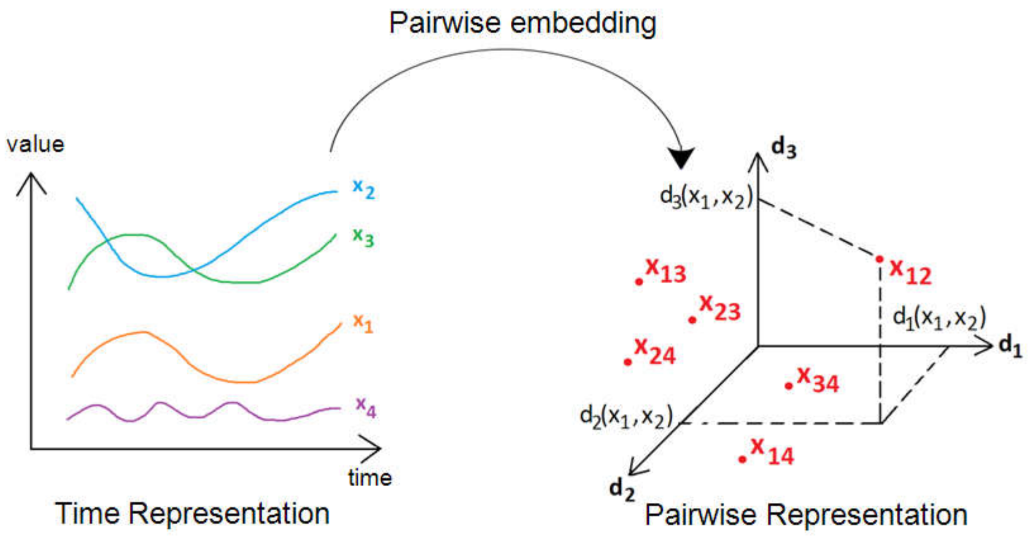


Figure 3.3: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the dissimilarity space (right). In this example, a pair of time series $(\mathbf{x}_1, \mathbf{x}_2)$ is projected into the dissimilarity space as a vector $\mathbf{x}_{12}$ described by $p = 3$ basic metrics: $\mathbf{x}_{12} = [d_1(\mathbf{x}_1, \mathbf{x}_2), d_2(\mathbf{x}_1, \mathbf{x}_2), d_3(\mathbf{x}_1, \mathbf{x}_2)]^T$.

A combination function D of the metrics d_h can be seen as a function in this space. In the following, we propose first to use a linear combination of d_h: $D(\mathbf{x}_i, \mathbf{x}_j) = \sum_h w_h.d_h(\mathbf{x}_i, \mathbf{x}_j)$. For simplification purpose, we denote $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_{ij})$ and the pairwise notation gives:

$$D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_{ij}) = \mathbf{w}^T \mathbf{x}_{ij} \tag{3.2}$$

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

3.2.2 Interpretation in the dissimilarity space

The interpretation of the data in the dissimilarity space is particular since the dissimilarity space is not a standard Euclidean space. The interpretation in this space requires to be careful.

If $\mathbf{x}_{ij} = \mathbf{0}$ then $\mathbf{x}_j$ is identical to $\mathbf{x}_i$ according to all metrics d_h. The norm of the 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 $\mathbf{x}_j$. Nevertheless, if two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ has their norms closed, it doesn't mean that the time series $\mathbf{x}_i$, $\mathbf{x}_j$, $\mathbf{x}_k$ and $\mathbf{x}_l$ are similar. Fig 3.4 shows an example of two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ that are close together in the dissimilarity 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 $\mathbf{x}_l$.

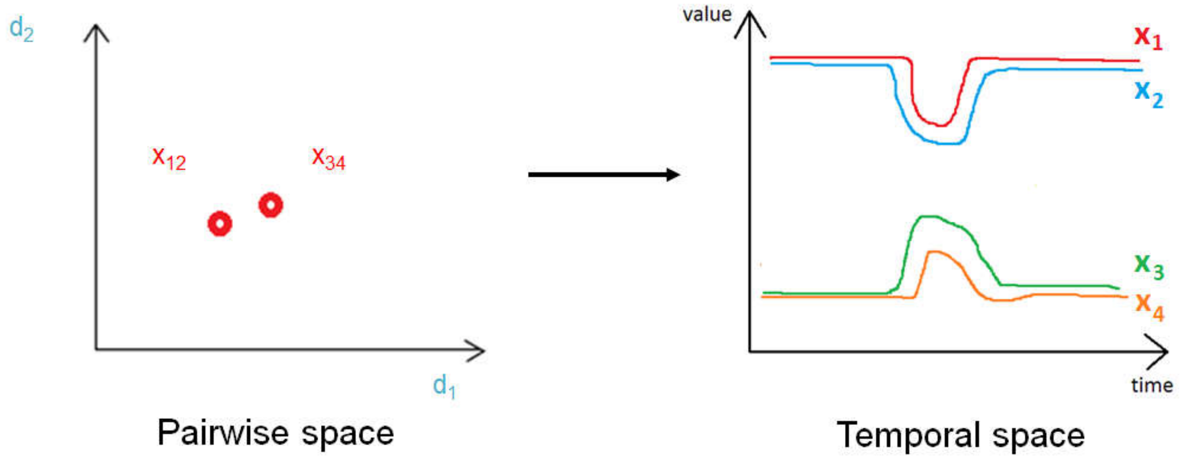


Figure 3.4: Example of two pairwise vectors $\mathbf{x}_{12}$ and $\mathbf{x}_{34}$ close in the dissimilarity space. However, the time series $\mathbf{x}_1$ and $\mathbf{x}_3$ are not similar in the temporal space.

A metric D that combines the p metrics $d_1, \ldots, d_p$ can be seen as a function of the dissimilarity space. It can be noticed that when the time series $\mathbf{x}_i$ are embedded in the pairwise, the information of their original class y_i is lost. Any multi-class problem is transformed in the dissimilarity space as a binary classification problem.

In the next sections, we transpose the metric learning problem for large margin nearest neighbor classifier in the dissimilarity space. We propose three formulations: Linear programming, Quadratic programming and SVM-based approach.

3.3 Linear Programming (LP) formalization

Let $\{\mathbf{x}_i, y_i\}_{i=1}^{n}$ be a set of n time series labeled y_i. We embed the n time series in the dissimilarity space and define $\{\mathbf{x}_{ij}, y_{ij}\}$ as the set of N pairwise vectors $\mathbf{x}_{ij}$ described by p metrics d_h and labeled $y_{ij} = +1$ if $y_j \neq y_i$ and -1 otherwise. Our objective is to define a metric D as a linear combination of the p metrics d_h (Eq. 3.2). In the dissimilarity space, the metric D should:

- **pull** to the origin the k nearest neighbors pairs $\mathbf{x}_{ij}$ of same labels $(y_{ij} = -1)$
- **push** from the origin all the pairs $\mathbf{x}_{il}$ of different classes $(y_{ij} = +1)$

Let $\mathbf{X}_{tar}$ be a $p \times (k.n)$ matrix containing all targets $\mathbf{x}_{ij}$. Fig. 3.5 illustrates our idea to learn the metric D. For each time series $\mathbf{x}_i$, we build the set of target pairs $\mathbf{x}_{ij}$ $(j \rightsquigarrow i)$ and the set of pairs $\mathbf{x}_{il}$ of different class. Then, we optimize the weight vector $\mathbf{w}$ so that the pairs $\mathbf{x}_{ij}$ are pulled to the origin and the pairs $\mathbf{x}_{il}$ are pushed from the origin.

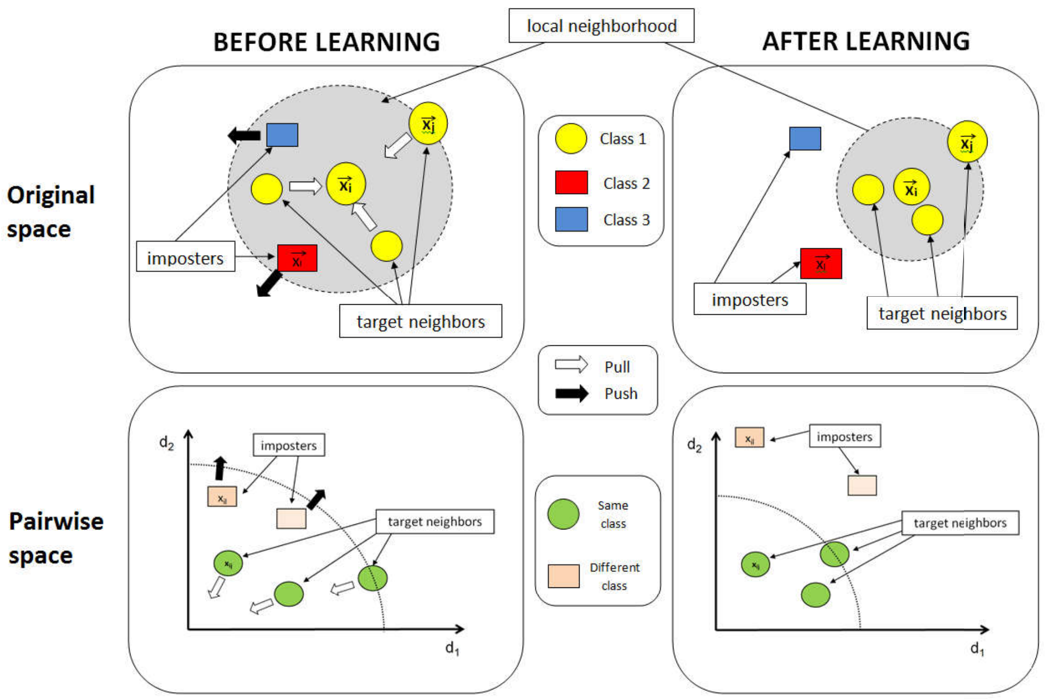


Figure 3.5: Geometric representation of the adaptation of metric learning problem from the original space (top) to the dissimilarity space (bottom) for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), imposters $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the 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).

Inspired from the Large Margin Nearest Neighbors (LMNN) framework proposed by Weinberger & Saul in Section 2.6.2, we transpose the metric learning problem into the dissimilarity space to learn a metric D that combines several unimodal metric d_h. In our problem, the optimal metric D is learned as the solution of a minimization problem, such that for each time series $\mathbf{x}_i$, it pulls its targets $\mathbf{x}_j$ and pushes all the samples $\mathbf{x}_l$ with a different label ($y_l \neq y_i$). The Metric Learning in Dissimilarity space (MLD) problem is formalized as:

$$\underset{D,\xi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{i,j \rightsquigarrow i} D(\mathbf{x}_{ij})}_{pull} + \underbrace{C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{ijl}}_{push} \right\} \tag{3.3}$$

s.t. $\forall j \rightsquigarrow i, l,$

$$D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.4}$$

$$\xi_{ijl} \geq 0 \tag{3.5}$$

where ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs. The proposed MLD differs from LMNN in which the push term in MLD considers all samples $\mathbf{x}_l$ with a different label from $\mathbf{x}_i$, whereas in LMNN, only the imposters are taken into consideration (those whose invade the target perimeter). Intuitively, this due to the fact that we do not want that samples $\mathbf{x}_l$ with a different class that were not at the beginning imposters, become imposters during the optimization process. By considering all the samples $\mathbf{x}_l$, we ensure that at each step of the optimization process, if a sample $\mathbf{x}_l$ becomes an imposter, then it will violate the constraints in Eq. 3.5 and thus, its slack variables ξ_{ijl} will be penalized in the objective function (Eq. 3.3) :

- 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 (Eq. 3.3).

- 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 (Eq. 3.3).

- If $D(\mathbf{x}_{il}) > D(\mathbf{x}_{ij}) + 1$, $\xi_{ijl} = 0$ and the slack variable has no effect in the objective function (Eq. 3.3).

By considering a linear combination of the unimodal distance d_h (Chapter 2): $D(\mathbf{x}_i, \mathbf{x}_j) = \sum_h w_h.d_h(\mathbf{x}_i, \mathbf{x}_j)$, optimizing the metric D is equivalent to optimizing the weight vector $\mathbf{w}$. Eqs. 3.3 and 3.4 leads to the MLD primal formulation:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \underbrace{||\mathbf{X}_{tar}^T\mathbf{w}||_1}_{pull} + C \underbrace{\sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2}\xi_{ijl}}_{push} \right\} \tag{3.6}$$

$$\text{s.t. } \forall j \rightsquigarrow i, l,$$

$$\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.7}$$

$$\xi_{ijl} \geq 0 \tag{3.8}$$

where $||\mathbf{X}_{tar}^T\mathbf{w}||_1 = \sum_{ij} \mathbf{w}^T\mathbf{x}_{ij}$ denotes the L_1-norm of the matrix $\mathbf{X}_{tar}^T\mathbf{w}$. Similarly to SVM, a L_1 or L_2 norm can be chosen. L_1 norm will privilege sparse solution of $\mathbf{w}$.
MLD can be seen as a large margin problem in the dissimilarity space and parallels can be done with SVM. The "pull" term acts as a regularizer which aims to minimize the norm of $\mathbf{w}$. Similarly to SVM, minimizing the norm of $\mathbf{w}$ is equivalent to maximizing the margin $\frac{1}{||\mathbf{w}||_2}$ between target pairs $\mathbf{x}_{ij}$ and pairs of different class $\mathbf{x}_{il}$.

3.4 Quadratic Programming (QP) formalization

The primal formulation of MLD (Eqs. 3.6, 3.7 and 3.8) supposed that the metric D is a linear combination of the metrics d_h. The primal formulation being similar to the one of SVM, it can be derived into its dual form to obtain non-linear solutions for D. For that, we consider in the objective function (Eq. 3.6), the square of the L_2-norm on $\mathbf{w}$ as the regularizer term, $\frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2$:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2}\xi_{ijl} \right\} \tag{3.9}$$

$$\text{s.t. } \forall j \rightsquigarrow i, l,$$

$$\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.10}$$

$$\xi_{ijl} \geq 0 \tag{3.11}$$

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 (Eq. 3.9) and the constraints (Eqs. 3.10 and 3.11) multiplied by their respective Lagrange multipliers $\boldsymbol{\alpha}$ and $\mathbf{r}$:

$$\begin{aligned} L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) =& \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C\sum_{ijl} \frac{1+y_{il}}{2}\xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\ & - \sum_{ijl} \alpha_{ijl}\left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl})\right) \end{aligned} \tag{3.12}$$

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:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{X}_{tar}^T\mathbf{X}_{tar}\mathbf{w} - \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) = 0$$

$$\frac{\partial L}{\partial \xi_{ijl}} = C - \alpha_{ijl} - r_{ijl} = 0$$

that leads to:

$$\mathbf{w} = (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1}\sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.13}$$

$$r_{ijl} = C - \alpha_{ijl} \tag{3.14}$$

Substituting Eq. 3.13 and 3.14 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.12, we get the MLD dual formulation[1]:

$$\underset{\boldsymbol{\alpha}}{\text{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'}(\mathbf{x}_{il} - \mathbf{x}_{ij})^T (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1}(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \tag{3.15}$$

s.t. $\forall\ i,\ j \rightsquigarrow i$ and l s.t. $y_{il} = +1$:

$$0 \leq \alpha_{ijl} \leq C \tag{3.16}$$

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

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T \mathbf{x}_{i'j'} \tag{3.17}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij})^T (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1} \mathbf{x}_{i'j'} \tag{3.18}$$

with $\mathbf{w}$ defined in Eq. 3.13. At the optimality, only the triplets $(\mathbf{x}_{il} - \mathbf{x}_{ij})$ with $\alpha_{ijl} > 0$ are considered as the support vectors. The direction $\mathbf{w}$ of the metric D is lead by these triplets. All other triplets have $\alpha_{ijl} = 0$ (non-support v ector), and the metric D is independent from this triplets. If we remove some of the non-support vectors, the metric D remains unaffected. From the viewpoint of optimization theory, we can also see this from the Karush-Kuhn-Tucker (KKT) conditions: the complete set of conditions which must be satisfied at the optimum of a constrained optimization problem. At the optimum, the Karush-Kuhn-Tucker (KKT) conditions apply, in particular:

$$\alpha_{ijl}(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) - 1 + \xi_{ijl}) = 0$$

from which we deduce that either $\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) > 1$ and $\alpha_{ijl} = 0$ (the triplet $(\mathbf{x}_{il} - \mathbf{x}_{ij})$ is a non-support vector), or $\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) = 1 - \xi_{ijl}$ and $\alpha_{ijl} > 0$ (the triplet is a support vector). Therefore, the learned metric D is a combination of scalar products between new pairs $\mathbf{x}_{i'j'}$ and a few number of triplets $\mathbf{x}_{ijl}$ of the training set.

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. 3.15 only relies on the inner product $(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'})^T (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1}(\mathbf{x}_{il} - \mathbf{x}_{ij})$. We can hence apply the kernel trick on Eqs. 3.17 and 3.18 to find non-linear solutions for D:

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T \phi(\mathbf{x}_{i'j'})$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl} \phi(\mathbf{x}_{il} - \mathbf{x}_{ij}) \phi(\mathbf{x}_{i'j'})$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl} K(\mathbf{x}_{il} - \mathbf{x}_{ij}; \mathbf{x}_{i'j'})$$

These equations suppose that the null vector $\mathbf{0}$ in the original space is transformed through the

[1]complete details of the calculations in Appendix D

transformation ϕ into the null vector: $\phi(\mathbf{0}) = \mathbf{0}$ in the feature space. We recall that $D(\mathbf{x}_{ii} = \mathbf{0})$ is expected to be equal to zero (distinguishability property in Section 2.2). However, if the vectors $\mathbf{x}_{ij}$ are projected in a feature space by a transformation ϕ, it doesn't guarantee that $\phi(\mathbf{0}) = \mathbf{0}$. Fig. 3.6 illustrates the idea for a polynomial kernel in which $\phi(\mathbf{0}) = [0, 0, 0, 1]^T$. Thus, the metric measure needs to be computed in the feature space relatively to the projection of $\phi(\mathbf{0})$. This is done by adding a term $\mathbf{w}^T\phi(\mathbf{0})$ to Eqs. 3.17 and 3.18:

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T\phi(\mathbf{x}_{i'j'}) - \mathbf{w}^T\phi(\mathbf{0}) \tag{3.19}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}\phi(\mathbf{x}_{il} - \mathbf{x}_{ij})\phi(\mathbf{x}_{i'j'} - \mathbf{x}_{ij}) - \sum_{ijl} \alpha_{ijl}\phi(\mathbf{x}_{il} - \mathbf{x}_{ij})\phi(\mathbf{0} - \mathbf{x}_{ij}) \tag{3.20}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}K\left(\mathbf{x}_{il} - \mathbf{x}_{ij}; \mathbf{x}_{i'j'} - \mathbf{x}_{ij}\right) - \sum_{ijl} \alpha_{ijl}K\left(\mathbf{x}_{il} - \mathbf{x}_{ij}; \mathbf{0} - \mathbf{x}_{ij}\right) \tag{3.21}$$

where $\mathbf{0}$ denotes the null vector. The resulting metric D is made of two terms. The first one, $\mathbf{w}^T\phi(\mathbf{x}_{i'j'})$, is the metric measure for a new pair $\mathbf{x}_{i'j'}$. The second term, $\mathbf{w}^T\phi(\mathbf{0})$, adapts the metric measure relatively to the origin point.

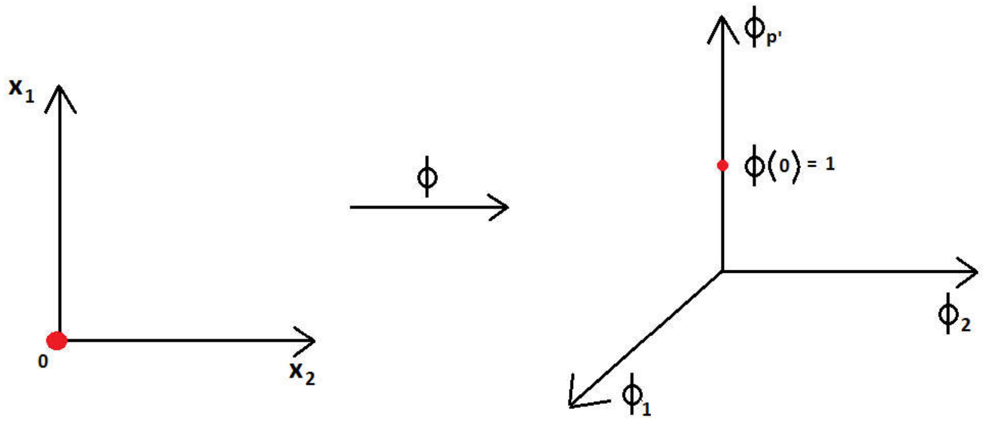


Figure 3.6: Illustration of samples in $\mathbb{R}^2$. The transformation ϕ for a polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + c)^d$ with $c = 1$ and $d = 2$ can be written explicitly: $\phi(\mathbf{x}_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, 1]^T$. The origin point $\mathbf{x}_i = [0, 0]^T$ is projected in the Hilbert space as $\phi(\mathbf{x}_i = \mathbf{0}) = [0, 0, 0, 1]^T$.

However, to define proper metrics that respects the properties of metrics (Section 2.2), specific kernels must be used. Our work don't propose any solutions to this problem but open the field for new research on this topic.

3.5 Support Vector Machine (SVM) approximation

3.5.1 Motivations

Many parallels have been studied between Large Margin Nearest Neighbors (LMNN) and SVM (Section 2.6.3). Similarly, the proposed MLD approach can be linked to SVM: both are convex optimization problem based on a regularized and a loss term. 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.

Motivated by these advantages, we propose to solve the MLD problem by solving a similar SVM problem. Then, we can naturally extend MLD approach to find non-linear solutions for the metric D thanks to the 'kernel trick'. In the following, we show the similarities and the differences between LP/QP and SVM formulation.

For a time series $\mathbf{x}_i$, we define the set of pairs $\mathbf{X}_{pi} = \{(\mathbf{x}_{ij}, y_{ij}) \text{ s.t. } j \rightsquigarrow i \text{ or } y_{ij} = +1\}$. It corresponds for a time series $\mathbf{x}_i$ to the set of pairs with target samples $\mathbf{x}_j$ (k nearest samples of same labels $j \rightsquigarrow i$) or samples $\mathbf{x}_l$ that has a different label from $\mathbf{x}_i$ ($y_l \neq y_i$) . Identity pairs $\mathbf{x}_{ii}$ are not considered. We refer to $\mathbf{X}_p = \bigcup\limits_i \mathbf{X}_{pi}$ and consider the following standard soft-margin weighted SVM problem on $\mathbf{X}_p$ [2]:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} + C \sum_{i,j,y_{ij}=+1} p_i^+ \xi_{ij} \right\} \\ &\text{s.t. } y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \end{aligned} \tag{3.22}$$

where p_i^- and p_i^+ are the weight factors for target pairs and pairs of different class.

We show in the following that solving the SVM problem in Eq. 3.22 for $\mathbf{w}$ and b solves a similar MLD problem in Eq. 3.9 for $D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(\mathbf{w}^T \mathbf{x}_{ij} + b)$. If we set p_i^+ being the half of the number of targets of $\mathbf{x}_i$ and p_i^-, the half of the number of time series L of a different class than $\mathbf{x}_i$:

$$p_i^+ = \frac{k}{2} = \sum_{j \rightsquigarrow i} \frac{1}{2} \tag{3.23}$$

$$p_i^- = \frac{L}{2} = \frac{1}{2} \sum_l \frac{1 + y_{il}}{2} \tag{3.24}$$

[2] the SVM formulation below divides the loss part into two terms similarly to asymetric SVM

3.5.2 Similarities and differences in the constraints

First, we recall the SVM constraints in Eq. 3.22:

$$y_{ij}(\mathbf{w}^T\mathbf{x}_{ij} + b) \geq 1 - \xi_{ij}$$

These constraints can be split into two sets of constraints:

$$\begin{aligned} -(\mathbf{w}^T\mathbf{x}_{ij} + b) &\geq 1 - \xi_{ij} \quad \text{(same class: } y_{ij} = -1) \\ (\mathbf{w}^T\mathbf{x}_{il} + b) &\geq 1 - \xi_{il} \quad \text{(different classes: } y_{ij} = +1) \end{aligned}$$

By defining $D(\mathbf{x}_{ij}) = \frac{1}{2}(\mathbf{w}^T\mathbf{x}_{ij} + b)$, this leads to:

$$\begin{aligned} -D(\mathbf{x}_{ij}) &\geq \frac{1}{2} - \frac{\xi_{ij}}{2} \\ D(\mathbf{x}_{il}) &\geq \frac{1}{2} - \frac{\xi_{il}}{2} \end{aligned}$$

By summing each constraint two by two, this set of constraints implies the following set of constraints:

$$\begin{cases} \bullet\forall i,j,k,l \text{ such that } y_{ij} = -1, \text{ and } y_{kl} = +1, i \neq j \text{ and } i \neq k : \\ D(\mathbf{x}_k, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{kl}+\xi_{ij}}{2} \\ \bullet\forall i,j,l \text{ such that } y_{ij} = -1, \text{ and } y_{il} = +1, i \neq j : \\ D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{il}+\xi_{ij}}{2} \end{cases} \tag{3.25}$$

By defining $\xi_{ijl} = \frac{\xi_{ij}+\xi_{il}}{2}$, the second constraint in Eq. 3.25 from the MDL formulation is the same as the constraints in the SVM formulation in Eq. 3.10.

However, an additional set of constraints is present in the SVM formulation (first set of constraints in Eq. 3.25) and not in the proposed MLD. Geometrically, this can be interpreted as superposing the neighborhoods of all samples $\mathbf{x}_i$, making the union of all of their target sets $\mathbf{X}_{pi}$, and then pushing away all imposters $\mathbf{x}_{il}$ from this resulting target set. This is therefore creating "artificial imposters" $\mathbf{x}_{kl}$ that don't violate the local target space of sample $\mathbf{x}_k$, but are still considered as imposters because they invade the target of sample $\mathbf{x}_i$ (because of the neighborhoods superposition) (Figure 3.7). This is more constraining in the SVM resolution for the resulting metric D especially if the neighborhoods have different spread.

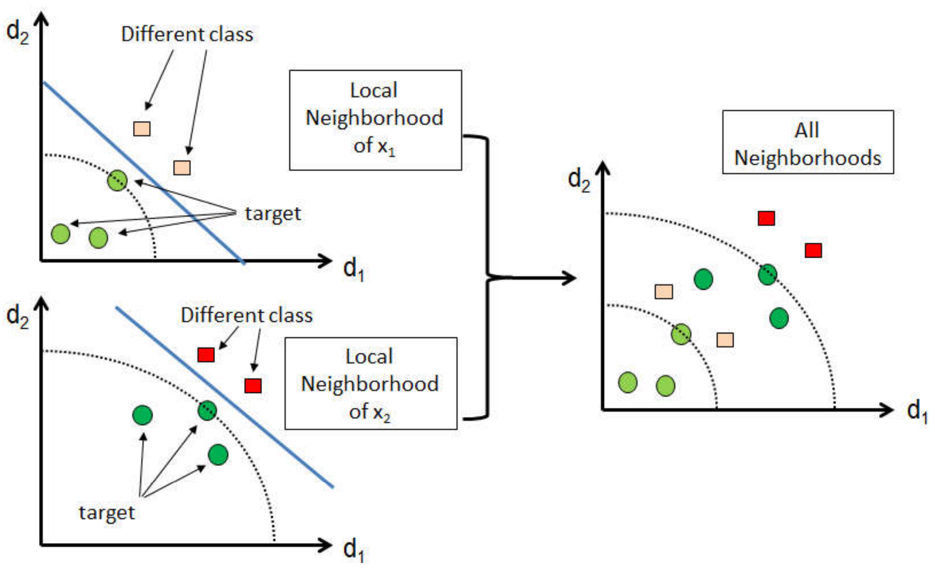


Figure 3.7: Geometric representation of the neighborhood of $k = 3$ for two time series $\mathbf{x}_1$ and $\mathbf{x}_2$ (left). For each neighborhood, time series of different class are represented by a square and the margin by a blue line. Taking each neighborhood separately, the problem is linearly separable (LP/QP formulation). By combining the two neighborhoods (SVM formulation), the problem is no more linearly separable and in this example, the time series of different class of $\mathbf{x}_1$ (orange square) are "artificial imposters" of $\mathbf{x}_2$.

3.5.3 Similarities and differences in the objective function

Mathematically, from Eq. 3.23, we write:

$$\begin{aligned}\sum_{i,l,y_{il}=+1} p_i^+ \xi_{il} &= \sum_{il} p_i^+ \frac{1+y_{il}}{2}\xi_{il}\\ &= \sum_{il}\left(\sum_{j\rightsquigarrow i}\frac{1}{2}\right)\frac{1+y_{il}}{2}\xi_{il}\\ &= \frac{1}{2}\sum_{i,j\rightsquigarrow i,l}\frac{1+y_{il}}{2}\xi_{il} \qquad (3.26)\end{aligned}$$

And from Eq. 3.24, we write:

$$\begin{aligned}\sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} &= \sum_{i,j\rightsquigarrow i} p_i^- \xi_{ij}\\ &= \sum_{i,j\rightsquigarrow i}\left(\frac{1}{2}\sum_{l}\frac{1+y_{il}}{2}\right)\xi_{ij}\\ &= \frac{1}{2}\sum_{i,j\rightsquigarrow i,l}\frac{1+y_{il}}{2}\xi_{ij} \qquad (3.27)\end{aligned}$$

By replacing Eqs. 3.26 and 3.27 back into Eq. 3.22, the objective function becomes:

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}\frac{\xi_{ij}+\xi_{il}}{2} \\ &\min_{\mathbf{w},\xi} \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w}}_{Regularization} + \underbrace{C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}\xi_{ijl}}_{Loss} \end{aligned} \tag{3.28}$$

Even if the loss part (push cost) is the same for both objective functions, the regularization part (pull cost) is different. In the SVM formulation (Eq. 3.28), the regularization part tends to minimize the norm of $\mathbf{w}$ whereas in MLD (Eq. 3.9), it tends to minimize the norm of $\mathbf{w}$ after a linear transformation through $\mathbf{X}_{tar}$. This transformation can be interpreted as a Mahalanobis norm in the dissimilarity space with $\mathbf{M} = \mathbf{X}_{tar}\mathbf{X}_{tar}^T$. Nevertheless, both have the same objective: improve the conditioning of the problem by enforcing solutions with small norms. In practice, even with these differences, the SVM provides suitable solutions for our time series metric learning problem.

3.5.4 Geometric interpretation

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. 3.9 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. 3.8. 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. 3.9 (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. 3.9 (right)), the problem involves all pairs and the margin is optimized so that pairs $\mathbf{x}_{ij}$ and $\mathbf{x}_{il}$ are globally separated.

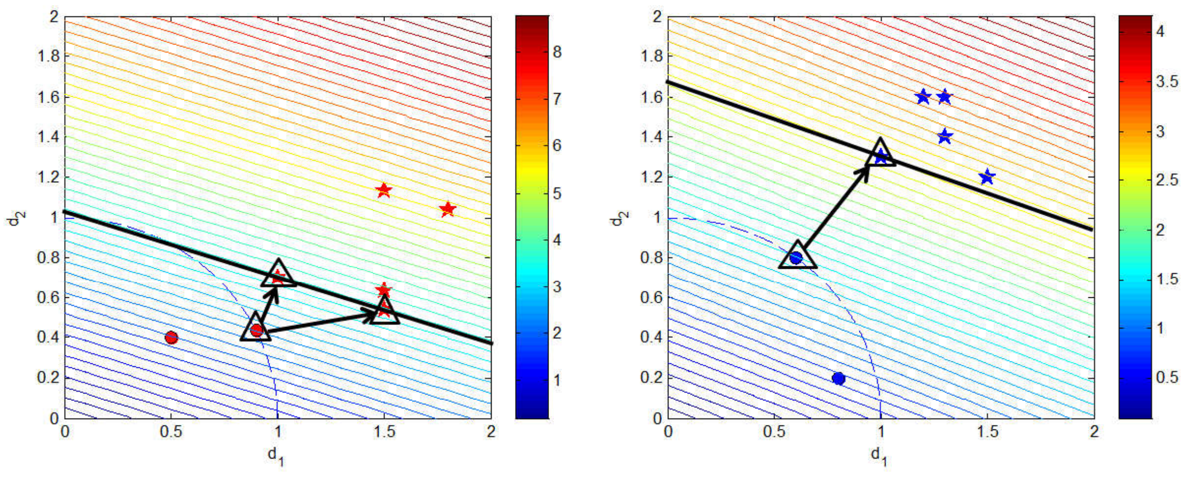


Figure 3.8: Solutions found by solving the LP problem for $k = 2$ neighborhood. Positive pairs (different classes) are indicated in stars and negative pairs (target pairs) are indicated in circle. Red and blue lines shows the margin when solving the problem for each neighborhood (red and blue points) 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.

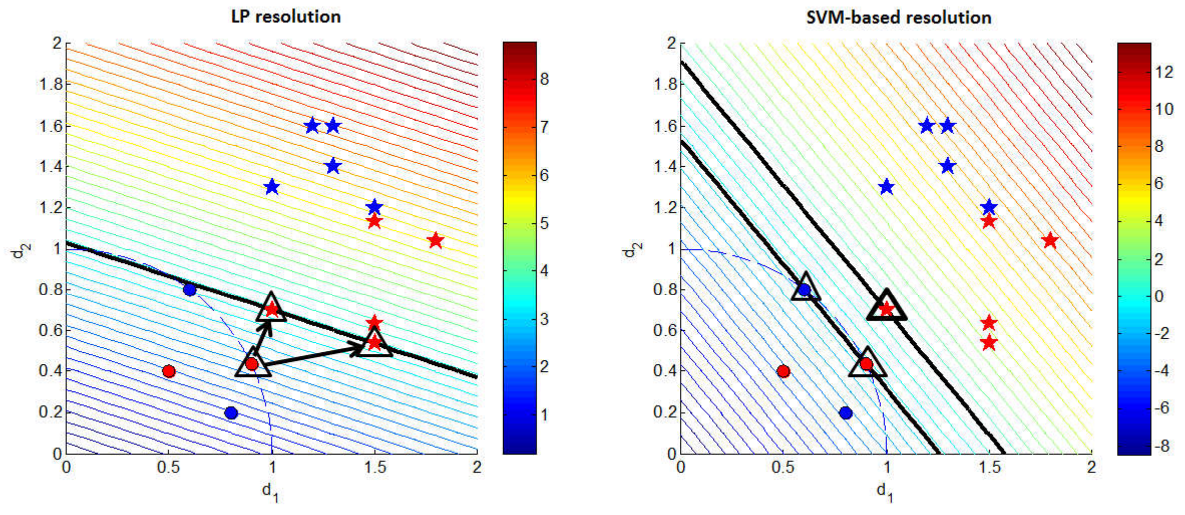


Figure 3.9: Solutions found by solving the LP problem (left) and the SVM problem (right). The global margin is indicated in black and the metric is represented in color levels. Support vectors made of triplets are indicated in black triangles. For the SVM, the black lines indicates the SVM canonical hyperplane where the support vector lies (black triangles).

3.6 Conclusion of the chapter

To learn a combined metric D from several unimodal metrics d_h that optimizes the k-NN performances, we first proposed a new space representation, the dissimilarity space where each pair of time series is projected as a vector described the unimodal metrics. Then, we propose three formalizations of our metric learning problem: Linear Programming, Quadratic Programming, SVM-based approximation. Table 3.1 sums up the main pros and cons of each formulation.

	LP	QP	SVM-based
Linear	Yes	Yes	Yes
Non-linear extension	No	Yes	Yes
Exact/Approximation resolution	Exact	Exact	Approximation
Sparcity	Yes	No	Yes/No

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

In the following, we consider the SVM-based approximation because SVM framework is well known and well implemented. In the next chapter, we give the details of the steps of our proposed algorithm: Multi-modal and Multi-scale Time series Metric Learning (M^2TML).

Chapter 4

Multi-modal and Multi-scale Time series Metric Learning ($\textsc{m}^2\textsc{tml}$) solution

Sommaire

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

4.1 Multi-scale approach

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

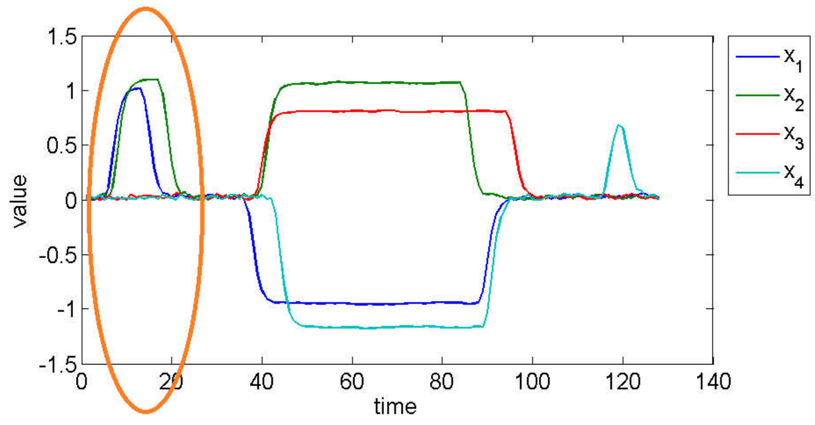


Figure 4.1: Example of 4 time series from the BME dataset, made of 3 classes : Begin, Middle and End. The 'Up' class has a characteristic bell at the beginning of the time series. The 'End' class has a characteristic bell at the end of the time series. The 'Middle' class has no characteristic bell. Orange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

Localizing patterns of interest in huge time series datasets has become an active area of search in many applications including diagnosis and monitoring of complex systems, biomedical data analysis, and data analysis in scientific and business time series . A large number of methods have been proposed covering the extraction of local features from temporal windows [BC94a] or the matching of queries according to a reference sequence [FRM94]. We focus on the computation of "local metrics".

ref

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

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,

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

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

$$I = [a; b] \tag{4.1}$$
$$I_L = [a; a + \alpha(b - a)] \tag{4.2}$$
$$I_R = [a - \alpha(b - a); b] \tag{4.3}$$

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

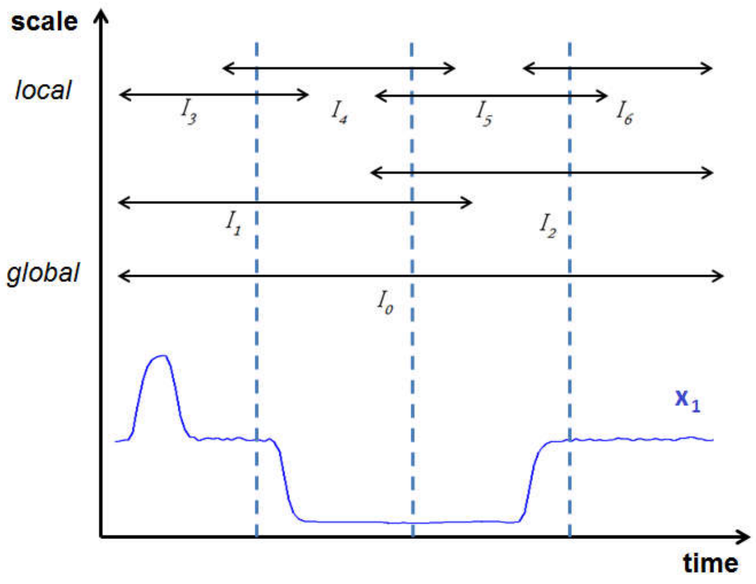


Figure 4.2: Multi-scale decomposition

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

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

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

4.2 Projection in the dissimilarity space

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

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

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

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

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

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

Pairwise space normalization
The scale between the p basic metrics d_h can be different. Thus, there is a need to scale the data within the pairwise space and ensure comparable ranges for the p basic metrics d_h. In our experiment, we use dissimilarity measures with values in $[0;+\infty[$. Therefore, we propose to Z-normalize their log distributions as explained in Section 1.1.4.

4.3 Neighborhood construction and scaling

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

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

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

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

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

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

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

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

The k nearest neighbors of a sample $\mathbf{x}_i$, denoted $\mathbf{x}_j$ ($j \rightsquigarrow i$), are defined in the pairwise space $\mathscr{E}$ by the k-th lowest norm $||\mathbf{x}_{ij}||_2$ negative pairs. Similarly, the **imposter set** X_i^{+*} is a subset of the positive set X_i^+ of pairs $\mathbf{x}_{il}$ such that the time series $\mathbf{x}_l$ is an imposter of $\mathbf{x}_i$, denoted $l \not\rightsquigarrow i$. 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:

$$X_i^{+*} = \{\mathbf{x}_{il}, y_{il} = +1\} \quad \text{s.t. } l \not\rightsquigarrow i \tag{4.11}$$

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

1. **k-NN vs impostors**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and impostor set:
$$X_p = \bigcup_i (X_i^{-*} \cup X_i^{+*}) \tag{4.12}$$
2. **k-NN vs all**: it corresponds to the union for all $\mathbf{x}_i$ of the target set and positive set. It ensures that no pairs $\mathbf{x}_{il}$ of different classes will invade the target neighborhood during the learning process:
$$X_p = \bigcup_i (X_i^{-*} \cup X_i^{+}) \tag{4.13}$$
3. **m-NN$^+$ vs m-NN$^-$**: it corresponds to the union for all $\mathbf{x}_i$ of the set of the m-nearest neighbors of the same class, denoted m-NN$_i^-$, and the m-nearest neighbor of $\mathbf{x}_i$ of a different class ($y_j \neq y_i$), denoted m-NN$_i^+$. For a k-NN classifier, by considering larger neighborhoods with $m = \alpha k (\alpha > 1)$ one includes more variability to generalize better the obtained solution:
$$X_p = \bigcup_i (m\text{-NN}_i^+ \cup m\text{-NN}_i^-) \tag{4.14}$$
   In the following, for simplification purpose, we define $m\text{-NN}^- = \bigcup_i m\text{-NN}_i^-$ as the union of all of the set of the m-nearest neighbors of the same class and $m\text{-NN}^- = \bigcup_i m\text{-NN}_i^+$ as the m-nearest neighbor of $\mathbf{x}_i$ of a different class.

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

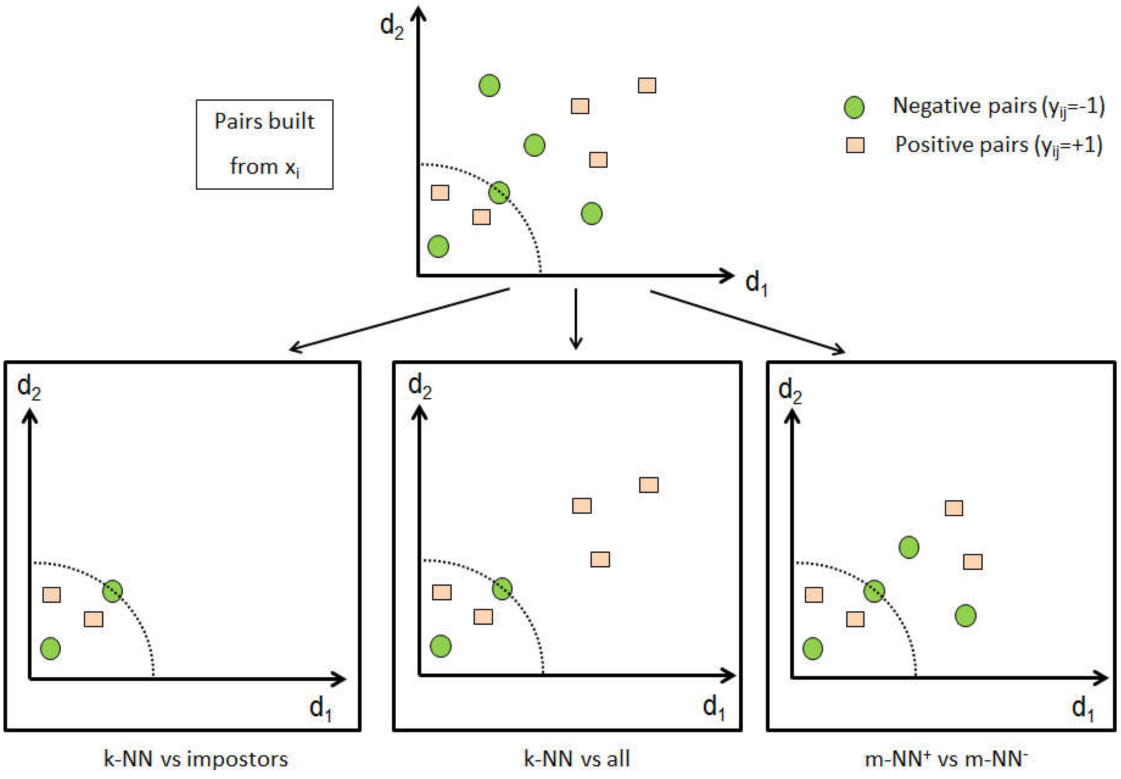


Figure 4.3: Example of a k-NN problem with $k = 2$. 3 different strategies (bottom) for pairwise training set X_p construction from the embedding of time series $\mathbf{x}_i$ in the pairwise space (top): k-NN vs impostor strategy (left), k-NN vs all strategy (middle) and m-NN$^+$ vs m-NN$^-$ (right)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

Neighborhood scaling

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

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

As explained in Chapter 3, Section 3.5.3, there exists an heterogeneity in the neighborhood. In real datasets, local neighborhoods can have very different scales as illustrated in Fig. 3.7. To make the target neighborhood spreads comparable, we propose for each $\mathbf{x}_i$ to scale its neighborhood vectors $\mathbf{x}_{ij}$ such that the L_2 norm (radius) of the farthest m-th nearest neighbor is 1:

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

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

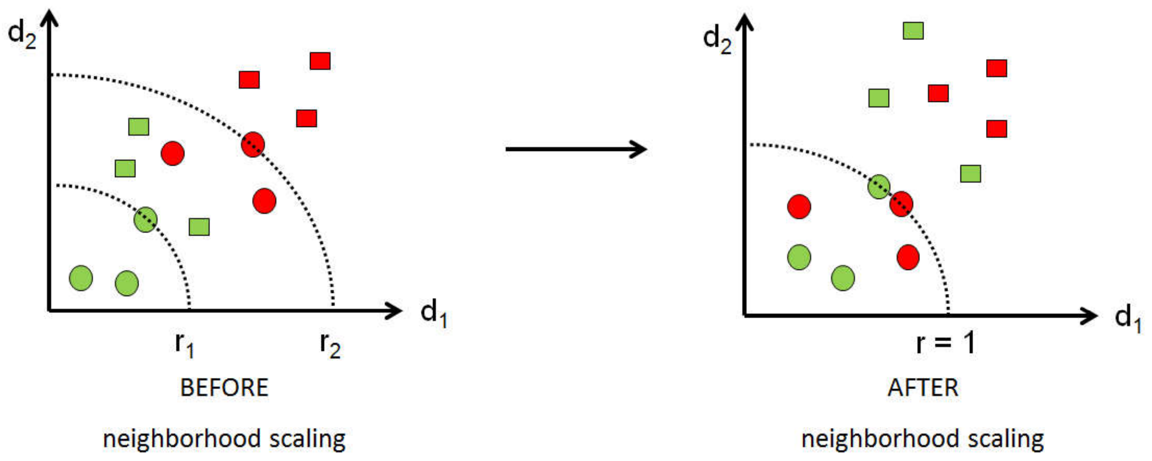


Figure 4.4: Effect of neighborhood scaling before (left) and after (right) on the neighborhood of two time series $\mathbf{x}_1$ (green) and $\mathbf{x}_2$ (red). Circle represent negative pairs (m-NN$^-$) and square represents positive pairs (m-NN$^+$) for $m = 2$ neighbors. Before scaling, the problem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

4.4 Definition of the dissimilarity measure

4.4.1 Support Vector Machine (SVM) resolution

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

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

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

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

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

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

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

4.4.2 Linear solutions

Let $\mathbf{x}_{test}$ be a new sample, $\mathbf{x}_{i,test} \in \mathscr{E}$ gives the proximity between $\mathbf{x}_i$ and $\mathbf{x}_{test}$ based on the p multi-modal and multi-scale metrics d_h. We denote $\mathbf{P}_\mathbf{w}(\mathbf{x}_{i,test})$ 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 positive and negative pairs. We review in this section different propositions to define the learned metric D: Scalar product, Projection Norm, Exponential transformation.

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

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

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

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

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

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

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

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

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

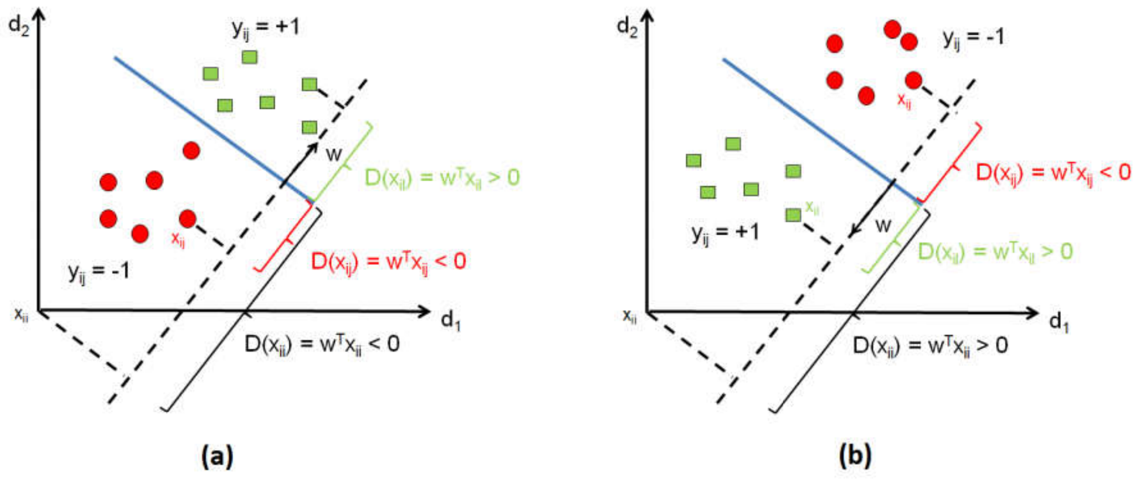


Figure 4.5: Example of SVM solutions and of the resulting metric D defined by a scalar product. Fig. (a) represents common expected configuration where negative pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin $\mathbf{x}_{ii} = 0$. In Fig. (b),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

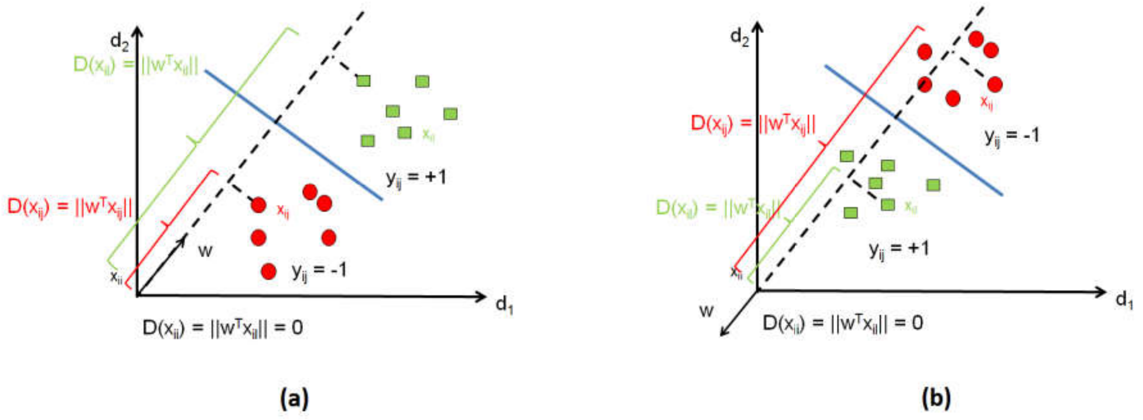
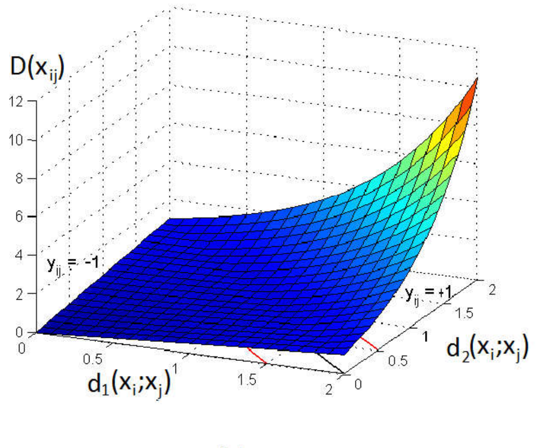


Figure 4.6: Example of SVM solutions and of the resulting metric D defined by the norm of the projection on $\mathbf{w}$. Fig. (a) represents common expected configuration where negative pairs ($\mathbf{x}_i$ and $\mathbf{x}_j$ are of same class) are situated in the same side as the origin $\mathbf{x}_{ii}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

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



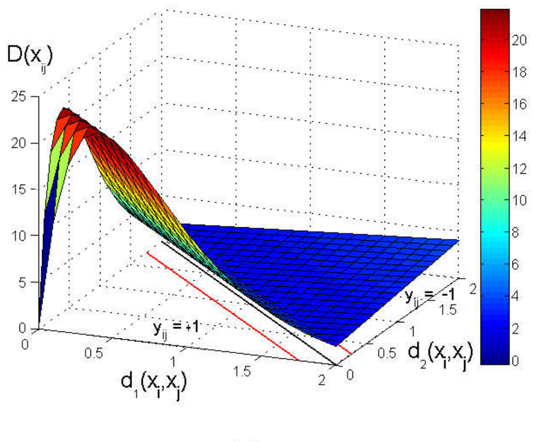


Figure 4.7: The behavior of the learned metric D ($p = 2$; $\lambda = 2.5$) with respect to common (a) and challenging (b) configurations of positive and negatives pairs.

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

4.4.3 Non-linear solutions

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

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

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

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

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

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

4.5 Algorithms and extensions

4.5.1 Algorithms

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

Algorithm 1 can be easily extended for multivariate and regression problem. First, for multivariate problem, each unimodal metric d_h can be computed for each variable. Then, the above framework can be applied. For regression problem, the label y_i for each time series $\mathbf{x}_i$ is a continuous value. The only modification is at the neighborhood steps, when defining the positive and negative pairs labeled y_{ij}. For that, in Chapter 3, Section 3.2.1, we propose two different strategies to define the pairwise labels y_{ij}.

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

```
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, 4.4
          the learned dissimilarity $D$ or $D_{\mathscr{H}}$ depending of the kernel $K$
2: Output: Predicted label $\hat{y}_{test}$
3: *Pairwise embedding*
   Embed pairs $(\mathbf{x}_i, \mathbf{x}_{test})$ $i \in 1, ..., N$ into $\mathscr{E}$ as described in Eq. 4.5 and normalize $d_h$s using
   the same normalization parameters in Algorithm 1
4: *Dissimilarity computation*
   Consider Eq. 4.23 (resp. Eq. 4.24) to compute $D(\mathbf{x}_i, \mathbf{x}_{test})$ (resp. $D_{\mathscr{H}}(\mathbf{x}_i, \mathbf{x}_{test})$) a linear
   (resp. non linear) combination function of the metrics $d_h(\mathbf{x}_i, \mathbf{x}_{test})$.
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 $\mathbf{x}_{test}$
```

4.5.2 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, the vector $\mathbf{x}_{ij}$ is labeled -1; and $+1$ otherwise."
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:

$$y_{ij} = \begin{cases} -1 & \text{if } y_i = y_j \\ +1 & \text{if } y_i \neq y_j \end{cases} \tag{4.26}$$

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 as illustrated in Fig. 4.8. 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}.

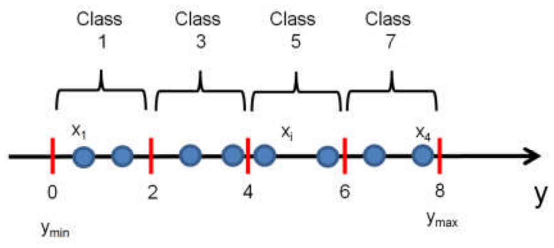


Figure 4.8: Example of discretization by binning a continuous label y into $Q = 4$ equal-length intervals. Each interval is associated to a unique class label. In this example, the class label for each interval is equal to the mean in each interval.

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, as illustrated in Fig 4.9. Moreover, a new sample $\mathbf{x}_j$ will have its labels y_j assigned to a class and not a real continuous value.

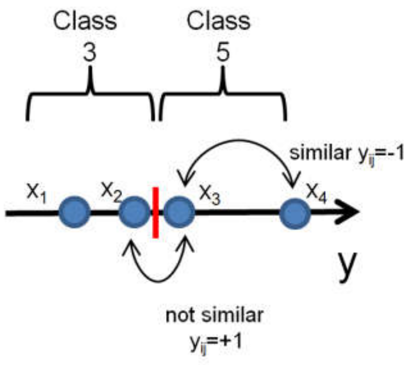


Figure 4.9: Border effect problems. In this example, $\mathbf{x}_2$ and $\mathbf{x}_3$ have closer value labels y_2 and y_3 than $\mathbf{x}_3$ and $\mathbf{x}_4$. However, with the discretization $\mathbf{x}_2$ and $\mathbf{x}_3$ don't belong to the same class and thus are consider as not similar.

The second approach considers the continuous value of y_i, computes a L_1-norm between the labels $|y_i - y_j|$ 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 between their labels $|y_i - y_j|$ is lower than ϵ (Fig. 4.10):

$$y_{ij} = \begin{cases} -1 & \text{if } |y_i - y_j| \leq \epsilon \\ +1 & \text{otherwise} \end{cases} \tag{4.27}$$

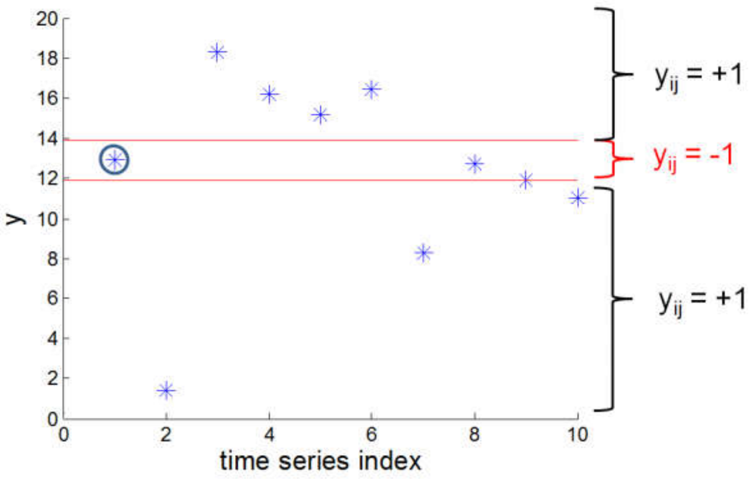


Figure 4.10: Example of pairwise label definition using an ϵ-tube (red lines) around the time series $\mathbf{x}_i$ (circled in blue). For, time series $\mathbf{x}_j$ that falls into the tube, the pairwise label is $y_{ij} = -1$ (similar) and outside of the tube, $y_{ij} = +1$ (not similar).

4.6 Conclusion of the chapter

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

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

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