CHAPTER 3

Pairwise space and Time series Metric Learning (TML) formalization

Sommaire

In this chapter, we formalize the problem of Time series Metric Learning (TML) which is the learning of a metric that combines several unimodal metrics for a robust k-NN classifier.
We first introduce a new space representation, the pairwise space. Secondly, we transpose the metric learning problem in the pairwise space. Finally, we propose three possible formulations: Linear programming, Quadratic programming and SVM-based approach.

3.1 Pairwise space representation

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 a new space representation, the pairwise space. Then, we present how to define pairwise labels for classification and regression problem. Finally, we give some interpretations in the pairwise space.

3.1.1 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 **pairwise space**. In this new space, illustrated in Fig. 3.1, 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.

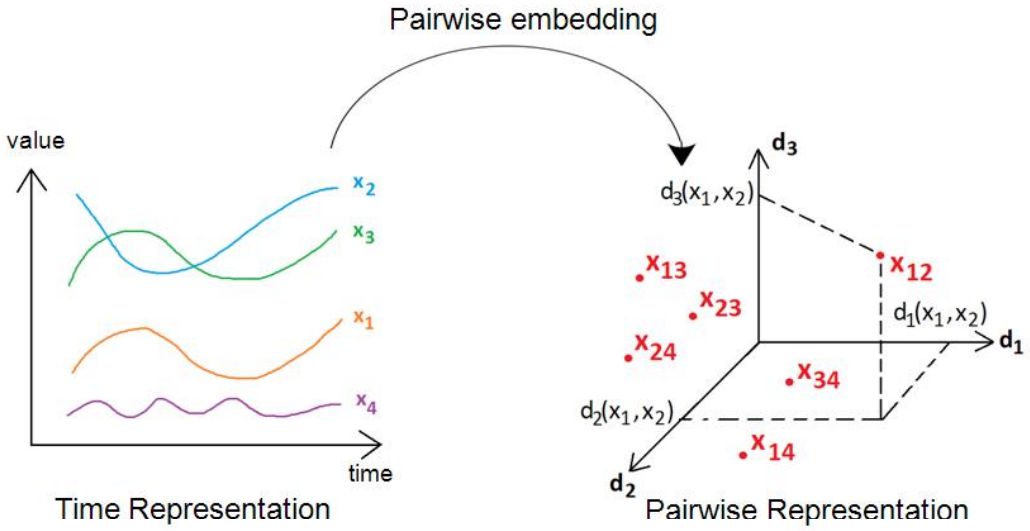

Figure 3.1: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the pairwise space (right). In this example, a pair of time series $(\mathbf{x}_1, \mathbf{x}_2)$ is projected into the pairwise space as a vector $\mathbf{x}_{12}$ described by $p = 3$ basic metrics: $\mathbf{x}_{12} = [d_1(\mathbf{x}_1, \mathbf{x}_2), d_2(\mathbf{x}_1, \mathbf{x}_2), d_3(\mathbf{x}_1, \mathbf{x}_2)]^T$.

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: $\mathbf{w} = [w_1, \ldots, w_p]^T$.

3.1.2 Pairwise label

In the pairwise 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.

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:

$$y_{ij} = \begin{cases} -1 & \text{if } y_i = y_j \\ +1 & \text{if } y_i \neq y_j \end{cases} \tag{3.3}$$

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. 3.2. 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}.

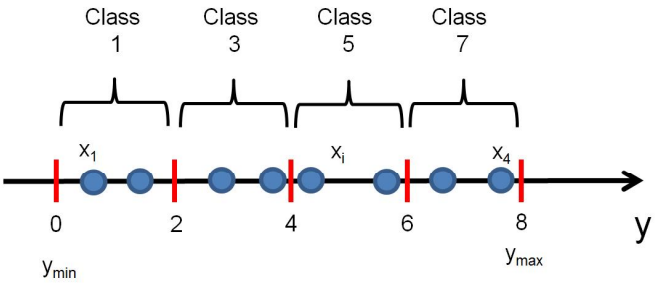

Figure 3.2: Example of discretization by binning a continuous label y into $Q = 4$ equal-length intervals. Each interval is associated to a unique class label. In this example, the class label for each interval is equal to the mean in each interval.

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

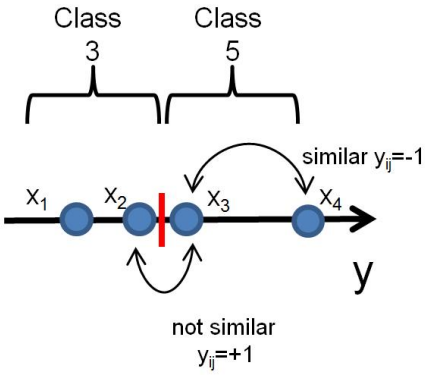

Figure 3.3: Border effect problems. In this example, $\mathbf{x}_2$ and $\mathbf{x}_3$ have closer value labels y_2 and y_3 than $\mathbf{x}_3$ and $\mathbf{x}_4$. However, with the discretization $\mathbf{x}_2$ and $\mathbf{x}_3$ don't belong to the same class and thus are consider as not similar.

The second approach considers the continuous value of y_i, computes a L1-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. 3.4):

$$y_{ij} = \begin{cases} -1 & \text{if } |y_i - y_j| \leq \epsilon \\ +1 & \text{otherwise} \end{cases} \tag{3.4}$$

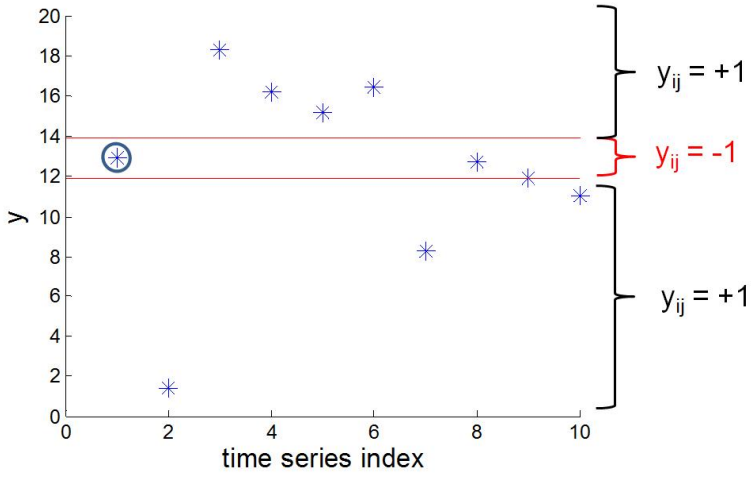

Figure 3.4: Example of pairwise label definition using an ϵ-tube (red lines) around the time series $\mathbf{x}_i$ (circled in blue). For, time series $\mathbf{x}_j$ that falls into the tube, the pairwise label is $y_{ij} = -1$ (similar) and outside of the tube, $y_{ij} = +1$ (not similar).

3.1.3 Interpretation in the pairwise space

The interpretation of the data in the pairwise space is particular since the pairwise 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 3.5 shows an example of two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ that are close together in the pairwise space. However, in the temporal space, the time series $\mathbf{x}_1$ and $\mathbf{x}_3$ are not similar for example. It means that $\mathbf{x}_i$ is as similar to $\mathbf{x}_j$ as $\mathbf{x}_k$ is to $\mathbf{x}_l$.

A metric D that combines the p unimodal metrics $d_1, \ldots, d_p$ can be seen as a function of the pairwise 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 pairwise space as a binary classification problem.

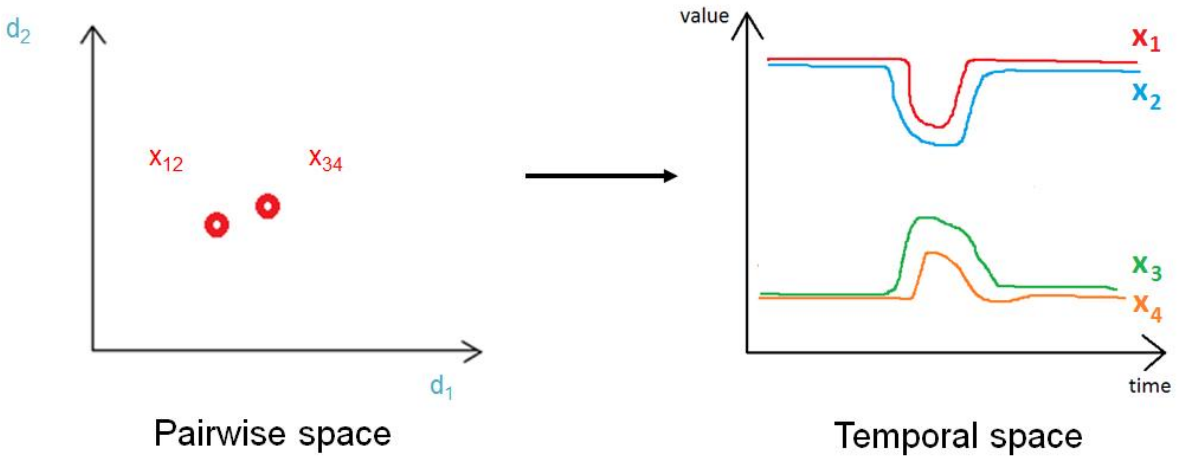

Figure 3.5: Example of two pairwise vectors $\mathbf{x}_{12}$ and $\mathbf{x}_{34}$ close in the pairwise space. However, the time series $\mathbf{x}_1$ and $\mathbf{x}_3$ are not similar in the temporal space.

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

3.2 Linear Programming (LP) formalization

Our objective is to define a metric D as a linear combination of the unimodal metric d_h (Eq. 3.2). In the pairwise 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 ($y_{il} = +1$)

Fig. 3.6 illustrates our idea. 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 ($y_{il} = +1$). Then, we optimise 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.

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 pairwise space to learn a temporal metric D combining 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 ($y_l \neq y_i$).

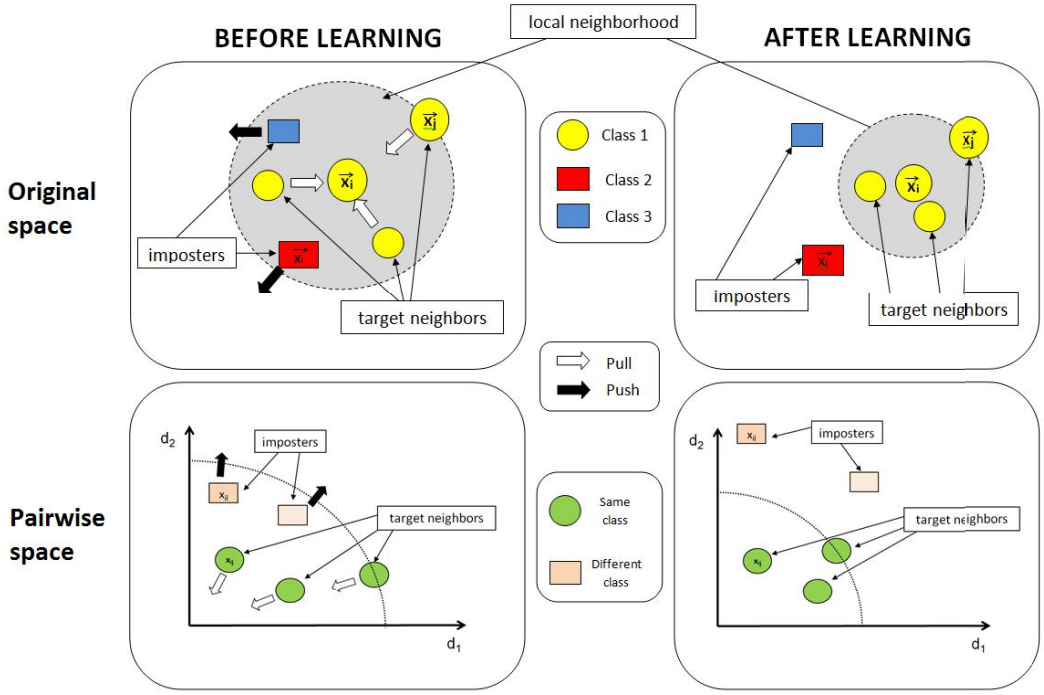

Figure 3.6: Geometric representation of the adaptation of metric learning problem from the original space (top) to the pairwise space (bottom) for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), imposters $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the pairwise 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).

The Time series Metric Learning (TML) problem in the pairwise space is formalized as:

$$\underset{D,\xi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{i,j \rightsquigarrow i} D(\mathbf{x}_{ij})}_{pull} + C \underbrace{\sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} . \xi_{ijl}}_{push} \right\} \tag{3.5}$$

$$\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i,$$

$$D(\mathbf{x}_{il}) - D(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.6}$$

$$\xi_{ijl} \geq 0 \tag{3.7}$$

where ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs. The proposed TML differs from LMNN in which the push term in TML 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. 3.7 and thus, its slack variables ξ_{ijl} will be penalized in the objective function (Eq. 3.5) :

- 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. 3.5).
- 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. 3.5).
- 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. 3.5).

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. 3.5 and 3.6 leads to the TML primal formulation:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \underbrace{||\mathbf{X}_{tar}^T\mathbf{w}||}_{pull} + C \underbrace{\sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl}}_{push} \right\} \tag{3.8}$$

$$\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i,$$

$$\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.9}$$

$$\xi_{ijl} \geq 0 \tag{3.10}$$

where $\mathbf{X}_{tar}$ is a $p \times (k.N)$ matrix containing all targets $\mathbf{x}_{ij}$ and $||\mathbf{X}_{tar}^T\mathbf{w}||$ denotes the norm of the vector $\mathbf{X}_{tar}^T\mathbf{w}$. As in SVM, a L1 or L2 norm can be chosen. L1 norm will priviledge sparse solution of $\mathbf{w}$.
TML can be seen as a large margin problem in the pairwise 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, mininimizing the norm of $\mathbf{w}$ is equivalent to maximizing the margin $\frac{1}{||\mathbf{w}||}$ between target pairs $\mathbf{x}_{ij}$ and pairs of different class $\mathbf{x}_{il}$.

3.3 Quadratic Programming (QP) formalization

The primal formulation of TML (Eqs. 3.8, 3.9 and 3.10) 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. 3.8), the square of the L2-norm on $\mathbf{w}$ as the regularizer term, $\frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2$:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl} \right\} \tag{3.11}$$

$$\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i,$$

$$\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \tag{3.12}$$

$$\xi_{ijl} \geq 0 \tag{3.13}$$

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. 3.11) and the constraints (Eqs. 3.12 and 3.13) multiplied by their respective Lagrange multipliers $\boldsymbol{\alpha}$ and $\mathbf{r}$:

$$\begin{aligned} L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) =& \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C\sum_{ijl} \frac{1+y_{il}}{2}\xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\ & - \sum_{ijl} \alpha_{ijl}\left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl})\right) \end{aligned} \tag{3.14}$$

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:

$$\mathbf{w} = (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1} \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.15}$$

$$r_{ijl} = C - \alpha_{ijl} \tag{3.16}$$

Substituting Eq. 3.15 and 3.16 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.14, we get the TML dual formulation[1]:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2}\sum_{ijl}\sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'}(\mathbf{x}_{il} - \mathbf{x}_{ij})^T(\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1}(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \tag{3.17}$$

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

$$0 \leq \alpha_{ijl} \leq C \tag{3.18}$$

[1] complete details of the calculations in Appendix D

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

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

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T (\mathbf{X}_{tar} \mathbf{X}_{tar}^T)^{-1} \mathbf{x}_{i'j'} \tag{3.20}$$

with $\mathbf{w}$ defined in Eq. 3.15. 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 vector), 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. 3.17 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. 3.19 and 3.20 to find non-linear solutions for D:

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

These equations suppose that the null vector $\mathbf{0}$ in the original space is transformed through the 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. 3.7 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. 3.19 and 3.20:

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T\phi(\mathbf{x}_{i'j'}) - \mathbf{w}^T\phi(\mathbf{0}) \tag{3.21}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl}\alpha_{ijl}\phi(\mathbf{x}_{il}-\mathbf{x}_{ij})\phi(\mathbf{x}_{i'j'}-\mathbf{x}_{ij}) - \sum_{ijl}\alpha_{ijl}\phi(\mathbf{x}_{il}-\mathbf{x}_{ij})\phi(\mathbf{0}-\mathbf{x}_{ij}) \tag{3.22}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl}\alpha_{ijl}K\left(\mathbf{x}_{il}-\mathbf{x}_{ij};\mathbf{x}_{i'j'}-\mathbf{x}_{ij}\right) - \sum_{ijl}\alpha_{ijl}K\left(\mathbf{x}_{il}-\mathbf{x}_{ij};\mathbf{0}-\mathbf{x}_{ij}\right) \tag{3.23}$$

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.

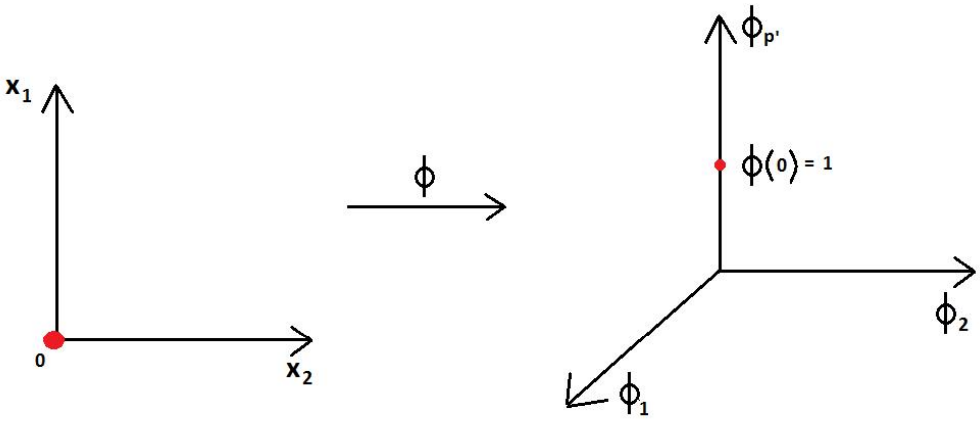

Figure 3.7: Illustration of samples in $\mathbb{R}^2$. The transformation ϕ for a polynomial kernel $K(\mathbf{x}_i,\mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + c)^d$ with $c = 1$ and $d = 2$ can be written explicitly: $\phi(\mathbf{x}_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, 1]^T$. The origin 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.

3.4 Support Vector Machine (SVM) approximation

3.4.1 Motivations

Many parallels have been studied between Large Margin Nearest Neighbors (LMNN) and SVM (Section 2.6.3). Similarly, the proposed TML 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 TML problem by solving a similar

SVM problem. Then, we can naturally extend TML 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 $\mathbf{X}_p = \bigcup_i \mathbf{X}_{pi}$ and consider the following standard soft-margin weighted SVM problem on $\mathbf{X}_p$ [2]:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} + C \sum_{i,j,y_{ij}=+1} p_i^+ \xi_{ij} \right\} \\ &\text{s.t. } y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \end{aligned} \tag{3.24}$$

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. 3.24 for $\mathbf{w}$ and b solves a similar TML problem in Eq. 3.11 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 $\mathbf{x}_i$:

$$p_i^+ = \frac{k}{2} = \sum_{j \rightsquigarrow i} \frac{1}{2} \tag{3.25}$$

$$p_i^- = \frac{L}{2} = \frac{1}{2} \sum_l \frac{1 + y_{il}}{2} \tag{3.26}$$

3.4.2 Similarities and differences in the constraints

First, we recall the SVM constraints in Eq. 3.24:

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

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:

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

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

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

$$\begin{cases} \bullet \forall i,j,k,l \text{ such that } y_{ij} = -1, \text{ and } y_{kl} = +1, i \neq j \text{ and } i \neq k : \\ D(\mathbf{x}_k, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{kl}+\xi_{ij}}{2} \\ \bullet \forall i,j,l \text{ such that } y_{ij} = -1, \text{ and } y_{il} = +1, i \neq j : \\ D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{il}+\xi_{ij}}{2} \end{cases} \tag{3.27}$$

By defining $\xi_{ijl} = \frac{\xi_{ij}+\xi_{il}}{2}$, the second constraint in Eq. 3.27 from the SVM formulation is the same as the constraints in the TML formulation in Eq. 3.12.

However, an additional set of constraints is present in the SVM formulation (first set of constraints in Eq. 3.27) and not in the proposed TML. 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 3.8). This is more constraining in the SVM resolution for the resulting metric D especially if the neighborhoods have different spread.

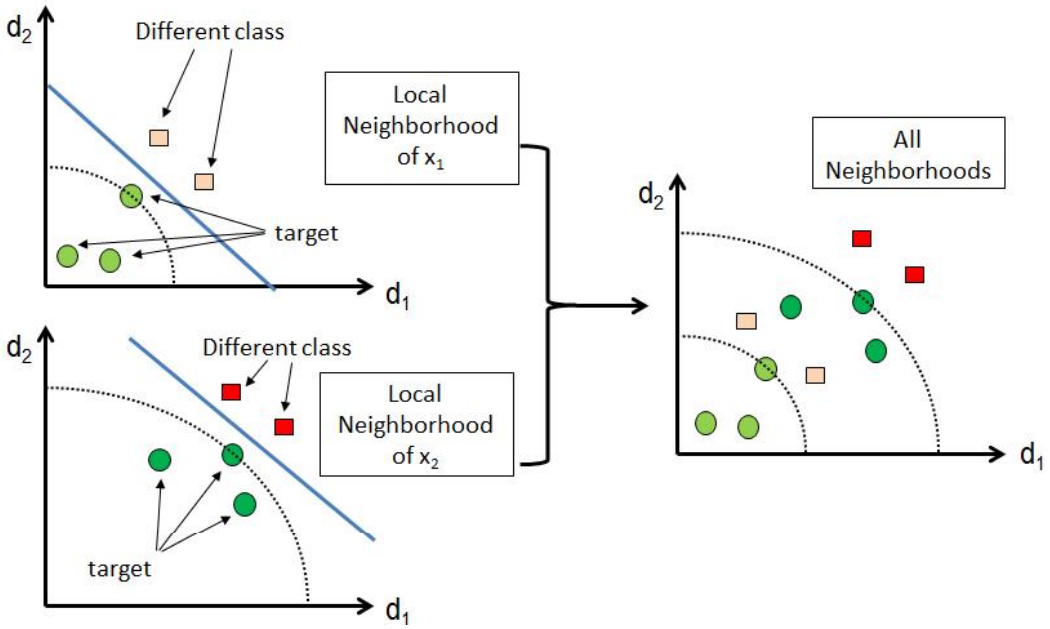

Figure 3.8: Geometric representation of the neighborhood of $k = 3$ for two time series $\mathbf{x}_1$ and $\mathbf{x}_2$ (left). For each neighborhood, time series of different class are represented by a square and the margin by a blue line. Taking each neighborhood separately, the problem is linearly 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 $\mathbf{x}_2$.

3.4.3 Similarities and differences in the objective function

Mathematically, from Eq. 3.25, we write:

$$\begin{aligned}\sum_{i,l,y_{il}=+1} p_i^+ \xi_{il} &= \sum_{il} p_i^+ \frac{1+y_{il}}{2} \xi_{il} \\ &= \sum_{il} \left(\sum_{j \rightsquigarrow i} \frac{1}{2} \right) \frac{1+y_{il}}{2} \xi_{il} \\ &= \frac{1}{2} \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{il} \end{aligned} \tag{3.28}$$

And from Eq. 3.26, we write:

$$\begin{aligned}\sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} &= \sum_{i,j \rightsquigarrow i} p_i^- \xi_{ij} \\ &= \sum_{i,j \rightsquigarrow i} \left(\frac{1}{2} \sum_{l} \frac{1+y_{il}}{2} \right) \xi_{ij} \\ &= \frac{1}{2} \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{ij} \end{aligned} \tag{3.29}$$

By replacing Eqs. 3.28 and 3.29 back into Eq. 3.24, the objective function becomes:

$$\begin{aligned}&\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \frac{\xi_{ij}+\xi_{il}}{2} \\ &\min_{\mathbf{w},\xi} \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w}}_{Regularization} + \underbrace{C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{ijl}}_{Loss} \end{aligned} \tag{3.30}$$

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. 3.30), the regularization part tends to minimize the norm of $\mathbf{w}$ whereas in TML (Eq. 3.11), 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 pairwise 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.

3.4.4 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. 3.10 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. 3.9. 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. 3.10 (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. 3.10 (right)), the problem involves all pairs and the margin is optimized so that pairs $\mathbf{x}_{ij}$ and $\mathbf{x}_{il}$ are globally separated.

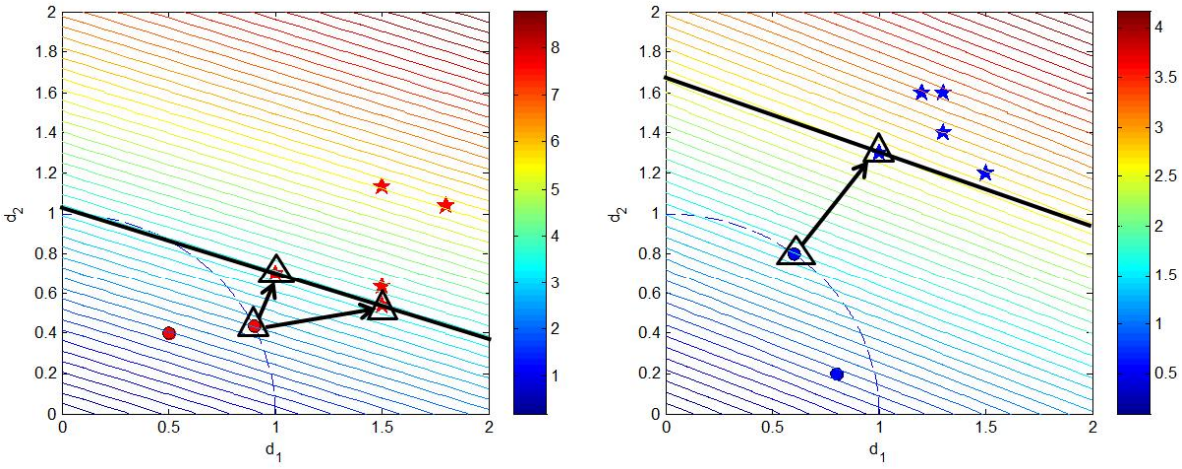

Figure 3.9: Solutions found by solving the LP problem for $k = 2$ neighborhood. Positive pairs (different classes) are indicated in stars and negative pairs (target pairs) are indicated in circle. Red and blue lines shows the margin when solving the problem for each neighborhood (red and blue points) 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.

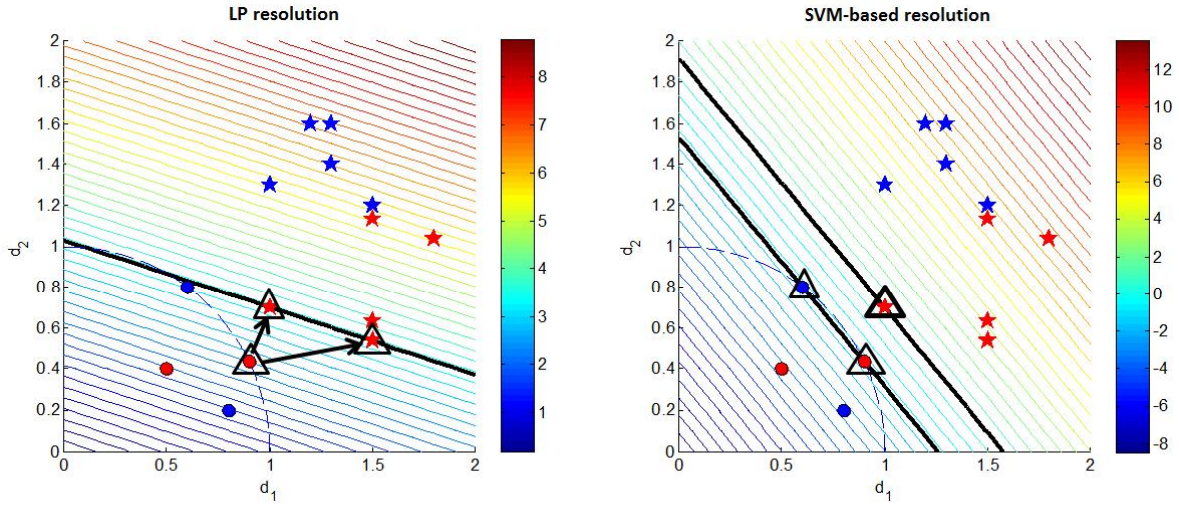

Figure 3.10: Solutions found by solving the LP problem (left) and the SVM problem (right). The global margin is indicated in black and the metric is represented in color levels. Support vectors made of triplets are indicated in black triangles. For the SVM, the black lines indicates the SVM canonical hyperplan where the support vector lies (black triangles).

3.5 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 pairwise 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.

In the following, we consider the SVM-based approximation because SVM framework is well known and well implementated. 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).