CHAPTER 3

Metric Learning in dissimilarity space: formalization

Sommaire

3.1 Motivations 53
3.2 Dissimilarity representation 56
3.3 Metric learning framework 57
3.4 Linear problem formalization 58
3.5 Non-linear formalization 60
3.6 Support Vector Machine (SVM) approximation 62
3.6.1 Motivations 62
3.6.2 Similarities and differences in the constraints 63
3.6.3 Similarities and differences in the objective function 64
3.6.4 Geometric interpretation 66
3.7 Conclusion of the chapter 67

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 or machine learning. Contrary to static data, temporal data are more complex: they may be compared not only on their amplitudes but also on their dynamic, frequential spectrum or other inherent characteristics. For time series comparison, a large number of metrics have been proposed, most of them are designed to capture similitudes and differences based on one temporal modality. For amplitude-based comparison, measures cover variants of Mahalanobis

distance or the dynamic time warping (DTW) to cope with delays [BC94b]; [Rab89]; [SC78b]; [KL83]. Other propositions refer to temporal correlations or derivative dynamic time warping for behavior-based comparison [AT10b]; [RBK08]; [CCP06]; [KP01]; [DM09]. For frequential aspects, comparisons are mostly based on the Discret Fourier or Wavelet Transforms [SS12a]; [KST98]; [DV10]; [Zha+06]. A detailed review of the major metrics is proposed in [MV14]. In general, the most discriminant modality (amplitude, behavior, frequency, etc.) varies from a dataset to another.

Furthermore, in some applications, the most discriminative characteristic between time series of different classes can be localized on a smaller part of the signal. A crucial key to localize discriminative features is to define metrics that involves totally or partially time series elements rather than systematically the whole elements. In the most challenging applications, it appears that both factors (modality, scale) are needed to discriminate the classes. Some works propose to combine several modalities through a priori models as in [DCDG10]; [DCA12]; [SB08].

Fig. 3.1 shows an example of significant improvement in classification performances by taking into account in the metric definition, several modalities (amplitude d_A, behavior d_B, frequential d_F) located at different scales (illustrated in the figure). The performance of the learnt combined metric is compared with the ones of the standard metrics that take into account for each, only one modality on a global scale (involving all time series elements).

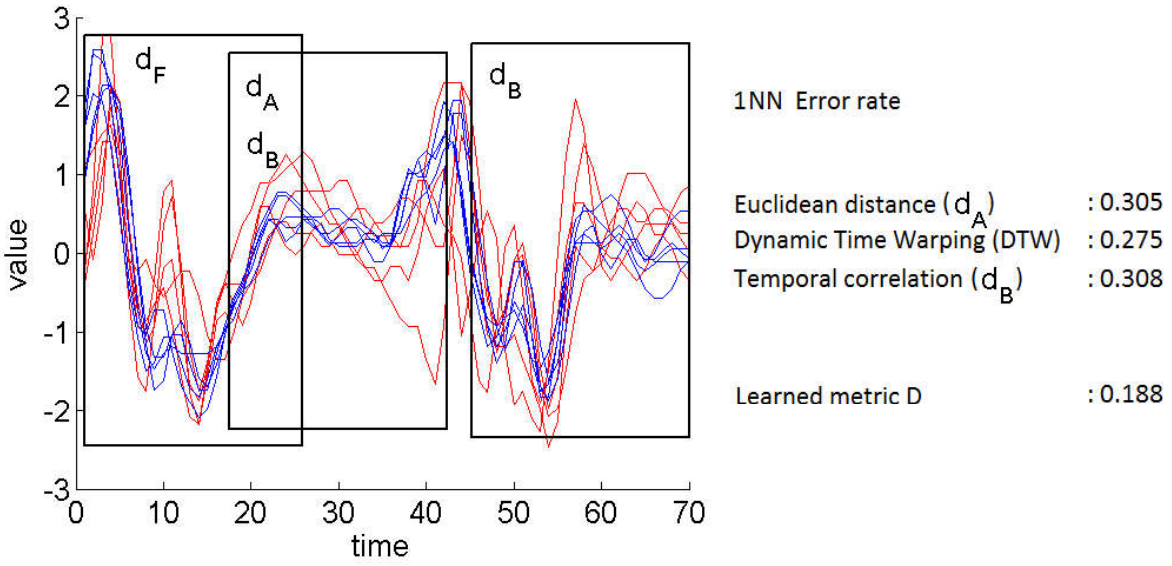

Figure 3.1: SonyAIBO dataset and error rate using a kNN ($k = 1$) with standard metrics (Euclidean distance, Dynamic Time Warping, temporal correlation) and a learned combined metric D. The figure shows the 4 major metrics involves in the combined metric D and their temporal scale (black rectangles).

Our aim is to take benefice of metric learning framework [WS09b]; [BHS12] to learn a multi-modal and multi-scale temporal metric for time series nearest neighbors classification. Specifically, our objective is to learn from the data a linear or non linear function that combines several temporal modalities at several temporal scales, that satisfies metric properties

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

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

In this chapter, we first present the concept of dissimilarity space. Then, we formalize the Metric Learning problem in the Dissimilarity space (MLD) and propose three formula-

tions: 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 detail the proposed solution to learn a multi-modal and multi-scale temporal metric (M^{2}TML) for a robust k-NN classifier.

3.2 Dissimilarity representation

Let $\{\mathbf{x}_i, y_i\}_{i=1}^n$ be a set of n time series labeled y_i. Let $d_1, ..., d_h..., d_p$ be p given metrics that allow to compare samples $\mathbf{x}_i$. For instance, in Chapter 2, we have proposed three types of metrics for time series: amplitude-based d_A, behavior-based d_B and frequential-based d_F. Our objective is to learn a metric $D = f(d_1, \ldots, d_p)$ that combines the p metrics in order to optimize the performance of a k-NN classifier.

The computation of a metric d, and D, always takes into account a pair of samples $(\mathbf{x}_i, \mathbf{x}_j)$. We introduce a new space representation referred as the **dissimilarity space**. In this new space, illustrated in Fig. 3.2, a vector $\mathbf{x}_{ij}$ represents a pair of time series $(\mathbf{x}_i, \mathbf{x}_j)$ described by the p metrics d_h: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$. We note n^2 the number of pairwise vectors $\mathbf{x}_{ij}$ induced by this embedding.

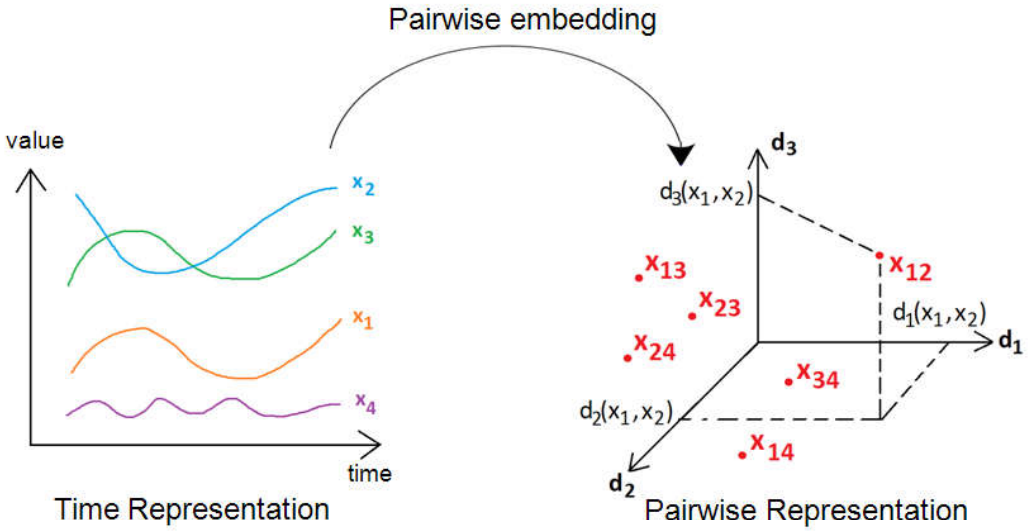

Figure 3.2: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the dissimilarity space (right) for $p = 3$ basic metrics.

In the dissimilarity space, a metric D that combines the p metrics $d_1, \ldots, d_p$ can be seen as a function of the dissimilarity space. The norm of a pairwise vector $||\mathbf{x}_{ij}||$ refers to the proximity between the time series $\mathbf{x}_i$ and $\mathbf{x}_j$. In particular, if $||\mathbf{x}_{ij}|| = 0$ then $\mathbf{x}_j$ is identical to $\mathbf{x}_i$ according to all metrics d_h. Note that a standard Euclidean distance between two pairwise vectors $\mathbf{x}_{ij}$ and $\mathbf{x}_{kl}$ in the dissimilarity space cannot give any interpretation about the proximity of the time series $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$ and $\mathbf{x}_l$ in the original space.

Comment [MR12]: figure enlevé ici

3.3 Metric learning framework

In this section, we formalize our problem. We propose to define the problem of learning a metric that combines several modality at several scales in the initial space as a metric learning problem.

Our objective is to learn a metric D that combines the p metrics: $D = f(d_1, \ldots, d_p)$. In addition, we want the metric D to optimize the performance of a k-NN classifier. It is based on two intuitions. First, for each sample $\mathbf{x}_i$, the metric D should bring closer the set its nearest neighbors $\mathbf{x}_j$, called $Pull_i$. Secondly, for each sample $\mathbf{x}_i$, the metric D should push the set of sample $\mathbf{x}_l$ that are not of the same class, called $Push_i$. Moreover, the metric D should verify the properties of a metric, *i.e.*, positivity, symmetry, distinguishability, triangular inequality. Formally, the metric learning problem can be written as the following optimization problem:

$$\begin{aligned} \underset{D,\xi}{\operatorname{argmin}} & \left\{ \underbrace{\sum_{\substack{i \\ j \in Pull_i}} D(\mathbf{x}_i, \mathbf{x}_j)}_{pull} + C \underbrace{\sum_{\substack{i \\ j \in Pull_i \\ l \in Push_i}} \frac{1 + y_{il}}{2} \xi_{ijl}}_{push} \right\} \\ \text{s.t. } & \forall j \rightsquigarrow i, l, \\ & D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijl} \\ & \xi_{ijl} \geq 0 \end{aligned} \tag{3.1}$$

where $y_{il} = +1$ if $y_i \neq y_l$ and -1 otherwise, ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs.

This formalization is similar to the one of Large Margin Nearest Neighbors (LMNN) proposed by Weinberger & Saul in [WS09a]. In LMNN, the set $Pull_i$ is defined as the k nearest neighbors $\mathbf{x}_j$ of the same class. The set $Push_i$ is defined by the sample $\mathbf{x}_l$ that invades the perimeter defined by the set $Pull_i$, called imposters. The set $Push_i$ being sparse, *i.e.*, only a subset of $\mathbf{x}_{il}$ is considered in the optimization process, the computational complexity is reduced. Thus, the problem is fast to solve. However, the sets $Pull_i$ and $Push_i$ are defined and fixed during the optimization process, according to an initial metric. If the initial metric is far from the solution, the optimum might not be attained. Also, note that in LMNN, $D(\mathbf{x}_i, \mathbf{x}_j)$ is replaced by $D^2(\mathbf{x}_i, \mathbf{x}_j)$.

We propose in our work to consider for the set $Pull_i$, a neighborhood larger than the ones of the k nearest neighbors, called the m neighborhood ($m \geq k$). We believe that considering a larger neighborhood during the training phase would improve the generalization properties of the learnt metric D during the testing phase. Similarly, we propose to consider for the set $Push_i$, samples that are in a scope larger than the ones of the set $Pull_i$, *i.e.*, considering samples $\mathbf{x}_l$ that are not only imposters. For example, we could consider all samples $\mathbf{x}_l$ that

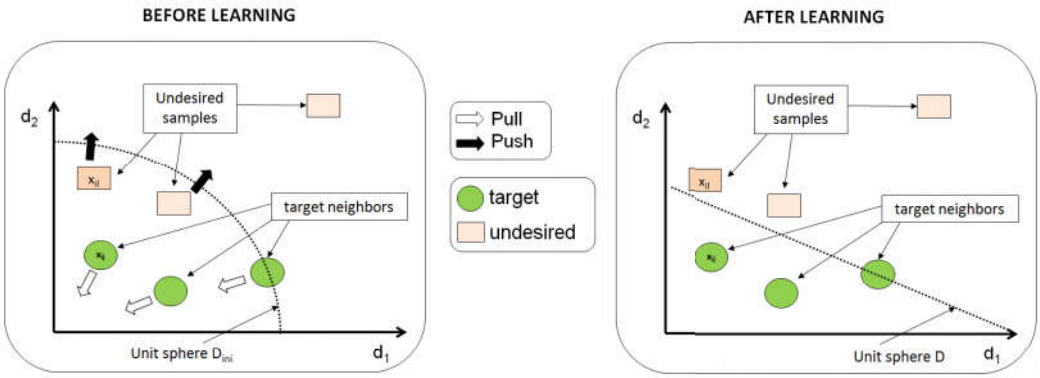

Figure 3.3: Geometric representation of the metric learning problem in the dissimilarity space for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), undesired samples $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the dissimilarity space, this is equivalent to have pairwise vectors $\mathbf{x}_{il}$ with a norm lower to some pairwise target $\mathbf{x}_{ij}$. The aim of metric learning is to push pairwise $\mathbf{x}_{il}$ (black arrow) and pull pairwise $\mathbf{x}_{ij}$ from the origin (white arrow).

have a different label from $\mathbf{x}_i$ ($y_l \neq y_i$). We call these samples $\mathbf{x}_l$ undesired samples. However, by considering all different samples, this would increase the computational cost of the optimization problem. An intermediate solution would consider for example the neighborhood of the m samples of different classes. More precisely, our solution states: $m = \alpha.k$ with $\alpha \geq 1$. Other propositions for m are also possible. Fig. 3.3 illustrates the concept.

In the next sections, we propose three formulations: Linear problem, Non-linear problem and SVM-based approximation.

3.4 Linear problem formalization

Let $\mathbf{X} = \{\mathbf{x}_{ij}, y_{ij}\}_{i,j=1}^{n}$ be a set of n^2 pairwise vectors $\mathbf{x}_{ij}$ described by p metrics: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$ labeled $y_{ij} = +1$ if $y_i \neq y_j$ and -1 otherwise. We consider a linear combination of the p metrics and use the pairwise notation for simplification purpose:

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

where $\mathbf{w}$ is the vector of weights w_h: $\mathbf{w} = [w_1, \ldots, w_p]^T$. We denote $\mathbf{W}$ the $p \times p$ matrix which diagonal elements are w_h and the other elements are zeros. We called $||\mathbf{W}\mathbf{x}_{ij}||_1 = \mathbf{w}^T \mathbf{x}_{ij}$ its L_1-norm. In that case, optimizing the metric D is equivalent to optimizing the weight vector $\mathbf{w}$. Eq. 3.1 leads to the primal formulation:

$$\begin{aligned}
&\underset{\mathbf{W},\xi}{\operatorname{argmin}} \left\{ \underbrace{\sum_{i} \sum_{j \in Pull_i} ||\mathbf{W}\mathbf{x}_{ij}||_1}_{pull} + C \underbrace{\sum_{i} \sum_{\substack{j \in Pull_i \\ l \in Push_i}} \frac{1+y_{il}}{2} \xi_{ijl}}_{push} \right\} \\
&\text{s.t. } \forall j \rightsquigarrow i, l, \\
&\quad \mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\
&\quad \xi_{ijl} \geq 0
\end{aligned} \tag{3.3}$$

where $y_{il} = +1$ if $y_i \neq y_l$ and -1 otherwise, ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs. We recall that the sets $Pull_i$ and $Push_i$ are defined as the m nearest neighbors of the same class and of different classes. From Eq. 3.3 , the slack variables ξ_{ijl} can be interpreted in the dissimilarity space as follow:

- If $D(\mathbf{x}_{il}) = \mathbf{w}^T\mathbf{x}_{il} < D(\mathbf{x}_{ij}) = \mathbf{w}^T\mathbf{x}_{ij}$, then the pairs $\mathbf{x}_{il}$ is an imposter pair that invades the neighborhood of the target pairs $\mathbf{x}_{ij}$. The slack variable $\xi_{ijl} > 1$ will be penalized in the objective function.
- If $D(\mathbf{x}_{il}) \geq D(\mathbf{x}_{ij})$ but $D(\mathbf{x}_{il}) \leq D(\mathbf{x}_{ij}) + 1$, then the pair $\mathbf{x}_{il}$ is within the safety margin of the target pairs $\mathbf{x}_{ij}$. The slack variable $\xi_{ijl} \in [0; 1]$ will have a small penalization effect in the objective function.
- If $D(\mathbf{x}_{il}) > D(\mathbf{x}_{ij}) + 1$, $\xi_{ijl} = 0$, then the slack variables ξ_{ijl} has no effect in the objective function. It corresponds to pairs $\mathbf{x}_{il}$ outside of the target neighborhood.

Also, the norm $||\mathbf{W}\mathbf{x}_{ij}||_1$ can be interpreted as a linear transformation $\mathbf{W}$ of the vectors $\mathbf{x}_{ij}$ in the dissimilarity space. Note that we aim at ensuring the properties of a metric for the learnt metric D. In particular, for positivity, for all $h = 1...p$, as d_h are dissimilarity measures ($d_h \geq 0$), it requires that :

$$w_h \geq 0 \tag{3.4}$$

This constraints is added up to the optimization problem in Eq. 3.3. Finally, it can be observed that Eq. 3.3 is a standard constraint optimization problem involving in the objective function, the sum of a regularized (pull cost in our case) R and a loss term L (push cost in our case). Formally:

$$\begin{aligned}
&\underset{\mathbf{W},\xi}{\operatorname{argmin}} \left[R(w) + L(\xi) \right] \\
&\text{s.t. } \textit{constaints}
\end{aligned} \tag{3.5}$$

Therefore, the problem could be extended by changing the regularizer R or loss term L. In the following, we extend the formulation to find non-linear solution for the metric D. For that, we propose to change the regularizer R, transform the optimization problem in Eq. 3.3 to obtain an equivalent optimization problem that depends only on an inner product.

3.5 Non-linear formalization

The formulation in Eq. 3.3 suppose 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 in Eq. 3.3, the square of the L_2-norm on $\mathbf{w}$ as the regularizer term, $\frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2$:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2}\xi_{ijl} \right\} \tag{3.6}$$

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

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

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

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. 3.6) and the constraints (Eqs. 3.7 and 3.8) multiplied by their respective Lagrange multipliers $\boldsymbol{\alpha}$ and $\mathbf{r}$:

$$\begin{aligned} L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r}) =& \frac{1}{2}||\mathbf{X}_{tar}^T\mathbf{w}||_2^2 + C\sum_{ijl}\frac{1+y_{il}}{2}\xi_{ijl} - \sum_{ijl} r_{ijl}\xi_{ijl} \\ & - \sum_{ijl} \alpha_{ijl}\left(\mathbf{w}^T(\mathbf{x}_{il} - \mathbf{x}_{ij} - 1 + \xi_{ijl})\right) \end{aligned} \tag{3.9}$$

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:

$$\mathbf{w} = (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1}\sum_{ijl}\alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij}) \tag{3.10}$$

$$r_{ijl} = C - \alpha_{ijl} \tag{3.11}$$

Substituting Eq. 3.10 and 3.11 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$ in Eq. 3.9, we get the MLD dual formulation[1]:

[1] complete details of the calculations in Appendix D

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \left\{ \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1} (\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \right\} \tag{3.12}$$

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

$$0 \leq \alpha_{ijl} \leq C \tag{3.13}$$

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

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T \mathbf{x}_{i'j'} \tag{3.14}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl} (\mathbf{x}_{il} - \mathbf{x}_{ij})^T (\mathbf{X}_{tar}\mathbf{X}_{tar}^T)^{-1} \mathbf{x}_{i'j'} \tag{3.15}$$

with $\mathbf{w}$ defined in Eq. 3.10. 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. 3.12 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. 3.14 and 3.15 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 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. 3.4 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. 3.14 and 3.15:

$$D(\mathbf{x}_{i'j'}) = \mathbf{w}^T\phi(\mathbf{x}_{i'j'}) - \mathbf{w}^T\phi(\mathbf{0}) \tag{3.16}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}\phi(\mathbf{x}_{il} - \mathbf{x}_{ij})\phi(\mathbf{x}_{i'j'} - \mathbf{x}_{ij}) - \sum_{ijl} \alpha_{ijl}\phi(\mathbf{x}_{il} - \mathbf{x}_{ij})\phi(\mathbf{0} - \mathbf{x}_{ij}) \tag{3.17}$$

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}K\left(\mathbf{x}_{il} - \mathbf{x}_{ij}; \mathbf{x}_{i'j'} - \mathbf{x}_{ij}\right) - \sum_{ijl} \alpha_{ijl}K\left(\mathbf{x}_{il} - \mathbf{x}_{ij}; \mathbf{0} - \mathbf{x}_{ij}\right) \tag{3.18}$$

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.

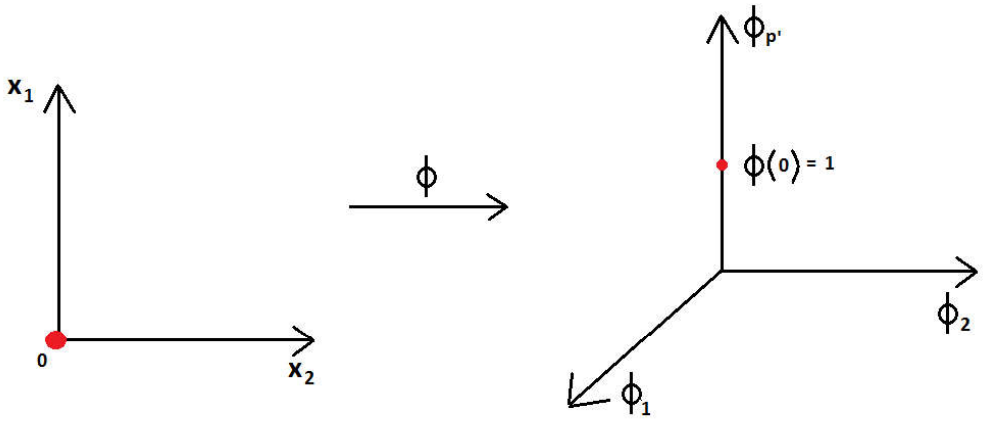

Figure 3.4: Illustration of samples in $\mathbb{R}^2$. The transformation ϕ for a polynomial kernel $K(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i^T\mathbf{x}_j + c)^d$ with $c = 1$ and $d = 2$ can be written explicitly: $\phi(\mathbf{x}_i) = [x_{i1}^2, x_{i2}^2, \sqrt{2}x_{i1}x_{i2}, 1]^T$. The origin 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.

3.6 Support Vector Machine (SVM) approximation

3.6.1 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 $\mathbf{X}_p = \bigcup_i \mathbf{X}_{pi}$ and consider the following standard soft-margin weighted SVM problem on $\mathbf{X}_p$ [2]:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{\operatorname{argmin}} \left\{ \frac{1}{2}||\mathbf{w}||_2^2 + C \sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} + C \sum_{i,j,y_{ij}=+1} p_i^+ \xi_{ij} \right\} \\ &\text{s.t. } y_{ij}(\mathbf{w}^T \mathbf{x}_{ij} + b) \geq 1 - \xi_{ij} \end{aligned} \tag{3.19}$$

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. 3.19 for $\mathbf{w}$ and b solves a similar MLD problem in Eq. 3.6 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 $\mathbf{x}_i$:

$$p_i^+ = \frac{k}{2} = \sum_{j \rightsquigarrow i} \frac{1}{2} \tag{3.20}$$

$$p_i^- = \frac{L}{2} = \frac{1}{2} \sum_l \frac{1 + y_{il}}{2} \tag{3.21}$$

3.6.2 Similarities and differences in the constraints

First, we recall the SVM constraints in Eq. 3.19:

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

These constraints can be split into two sets of constraints:

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

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

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

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

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

$$\begin{cases} \bullet \forall i,j,k,l \text{ such that } y_{ij} = -1, \text{ and } y_{kl} = +1, i \neq j \text{ and } i \neq k : \\ D(\mathbf{x}_k, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{kl}+\xi_{ij}}{2} \\ \bullet \forall i,j,l \text{ such that } y_{ij} = -1, \text{ and } y_{il} = +1, i \neq j : \\ D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{il}+\xi_{ij}}{2} \end{cases} \tag{3.22}$$

By defining $\xi_{ijl} = \frac{\xi_{ij}+\xi_{il}}{2}$, the second constraint in Eq. 3.22 from the MDL formulation is the same as the constraints in the SVM formulation in Eq. 3.7.

However, an additional set of constraints is present in the SVM formulation (first set of constraints in Eq. 3.22) 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 3.5). This is more constraining in the SVM resolution for the resulting metric D especially if the neighborhoods have different spread.

3.6.3 Similarities and differences in the objective function

Mathematically, from Eq. 3.20, we write:

$$\begin{aligned} \sum_{i,l,y_{il}=+1} p_i^+ \xi_{il} &= \sum_{il} p_i^+ \frac{1+y_{il}}{2} \xi_{il} \\ &= \sum_{il} \left(\sum_{j \rightsquigarrow i} \frac{1}{2} \right) \frac{1+y_{il}}{2} \xi_{il} \\ &= \frac{1}{2} \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{il} \end{aligned} \tag{3.23}$$

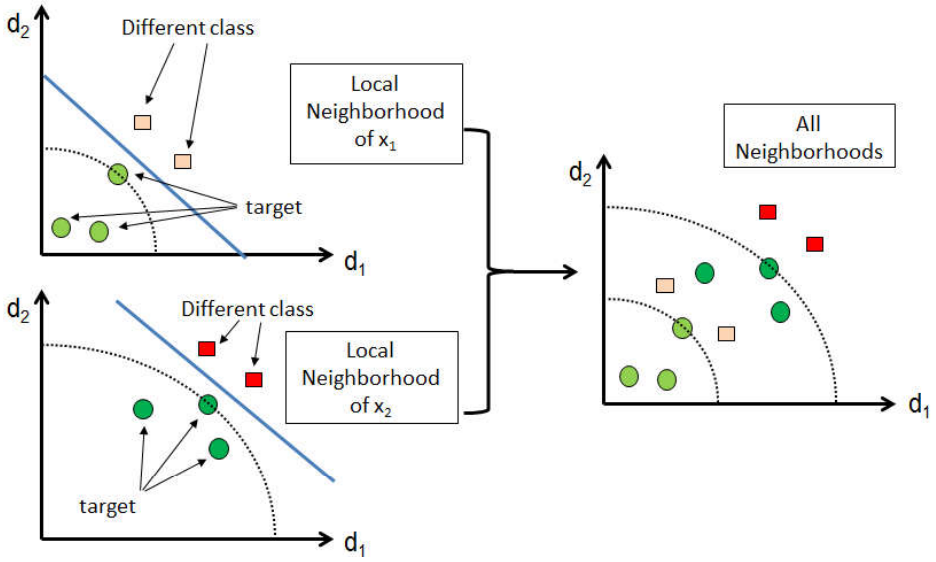

Figure 3.5: Geometric representation of the neighborhood of $k = 3$ for two time series $\mathbf{x}_1$ and $\mathbf{x}_2$ (left). For each neighborhood, time series of different class are represented by a square and the margin by a blue line. Taking each neighborhood separately, the problem is linearly 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 $\mathbf{x}_2$.

And from Eq. 3.21, we write:

$$\begin{aligned}\sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} &= \sum_{i,j\rightsquigarrow i} p_i^- \xi_{ij} \\ &= \sum_{i,j\rightsquigarrow i} \left(\frac{1}{2} \sum_{l} \frac{1+y_{il}}{2} \right) \xi_{ij} \\ &= \frac{1}{2} \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{ij} \end{aligned} \tag{3.24}$$

By replacing Eqs. 3.23 and 3.24 back into Eq. 3.19, the objective function becomes:

$$\begin{aligned} &\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}^T\mathbf{w} + C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2} \frac{\xi_{ij}+\xi_{il}}{2} \\ &\min_{\mathbf{w},\xi} \underbrace{\frac{1}{2}\mathbf{w}^T\mathbf{w}}_{Regularization} + \underbrace{C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2} \xi_{ijl}}_{Loss} \end{aligned} \tag{3.25}$$

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. 3.25), the regularization part tends

to minimize the norm of $\mathbf{w}$ whereas in MLD (Eq. 3.6), 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.

3.6.4 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. 3.7 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. 3.6. 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. 3.7 (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. 3.7 (right)), the problem involves all pairs and the margin is optimized so that pairs $\mathbf{x}_{ij}$ and $\mathbf{x}_{il}$ are globally separated.

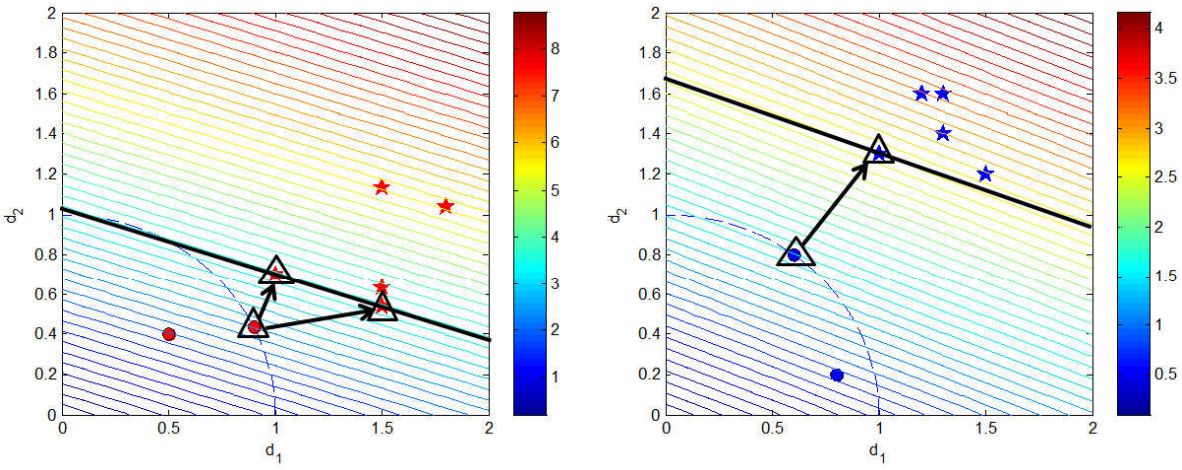

Figure 3.6: Solutions found by solving the LP problem for $k = 2$ neighborhood. Positive pairs (different classes) are indicated in stars and negative pairs (target pairs) are indicated in circle. Red and blue lines shows the margin when solving the problem for each neighborhood (red and blue points) 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.

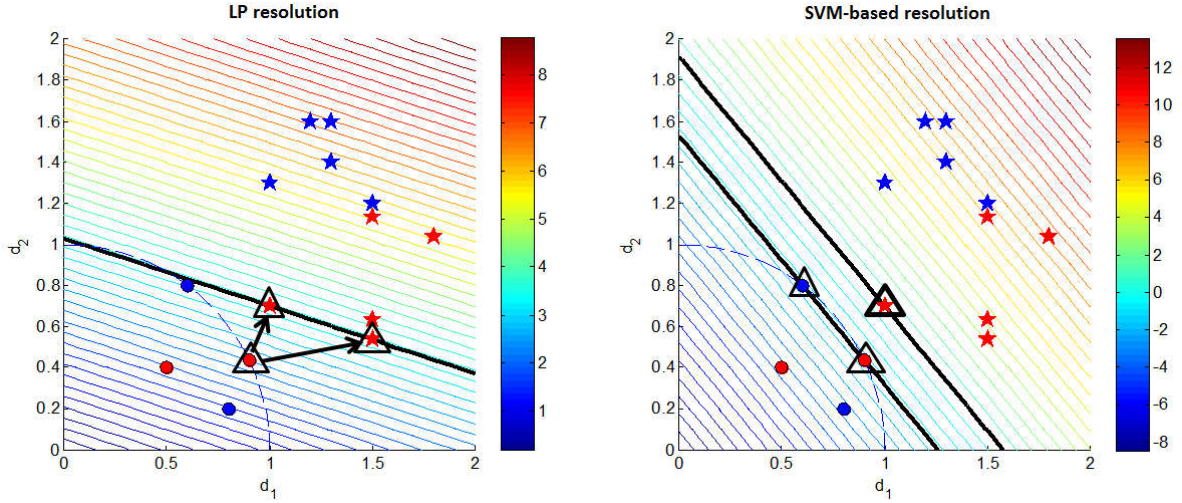

Figure 3.7: Solutions found by solving the LP problem (left) and the SVM problem (right). The global margin is indicated in black and the metric is represented in color levels. Support vectors made of triplets are indicated in black triangles. For the SVM, the black lines indicates the SVM canonical hyperplane where the support vector lies (black triangles).

3.7 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).