CHAPTER 2

Time series metrics and Metric Learning

Sommaire

In this chapter, we first present the definition of time series. Then, we recall the general properties of a metric and introduce some metrics proposed for time series. In particular, we focus on amplitude-based, behavior-based and frequential-based metrics. As real time series are subjected to varying delays, we recall the concept of alignment and dynamic programming. Then, we present some proposed combined metrics for time series. Finally, we review the concept of metric learning.

2.1 Definition of a time series

Time series are frequently data that can be found in various emerging applications such as sensor networks, smart buildings, social media networks or Internet of Things (IoT) [Naj+12]; [Ngu+12]; [YG08]. They are involved in many learning problems such as recognizing a human movement in a video, detect a particular operating mode, etc. [PAN+08]; [Ram+08].

In **clustering** problems, one would like to organize similar time series together into homogeneous groups. In **classification** problems, the aim is to assign time series to one of several predefined categories (e.g., different types of defaults in a machine). In **regression** problems, the objective is to predict a continuous value from observed time series (e.g., forecasting the measurement of a power meter from pressure and temperature sensors). Due to their temporal and structured nature, time series constitute complex data to be analyzed by classic machine learning approaches.

For physical systems, a time series of length T can be seen as a signal, sampled at a frequency f_e, in a temporal window $[0; \frac{T}{f_e}]$. From a mathematical perspective, a time series is a collection of a finite number of normalized observations made sequentially at discrete time instants $t = 1, ..., Q$. Note that when $f_e = 1$, $Q = T$.

Let $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iQ})$ be a univariate time series of length Q. Each observation x_{it} is bounded (i.e., the infinity is not a valid value: $x_{it} \neq \pm\infty$). The time series $\mathbf{x}_i$ is said to be univariate if the collection of observations x_{it} comes from the observations of one variable (i.e., the temperature measured by one sensor). When the observations are made at the same time from p variables (several sensors such as the temperature, the pressure, etc.), the time series is said multivariate. One possible representation is $\mathbf{x}_i = (\mathbf{x}_{i,1}, ..., \mathbf{x}_{i,p}) = (x_{i1,1}, ..., x_{iQ,1}, x_{i1,2}, ..., x_{i1,p}, ..., x_{iQ,p})$, where $\mathbf{x}_{i,j} = (\mathbf{x}_{i1,j}, \ldots, \mathbf{x}_{iQ,j})$. For simplification purpose, we consider in the following univariate time series.

Some authors propose to extract representative features from time series. Fig. 2.1 illustrates a model for time series proposed by Chatfield in [Cha04]. It states that a time series can be decomposed into 3 components: a trend, a cycle (periodic component) and a residual (irregular variations).

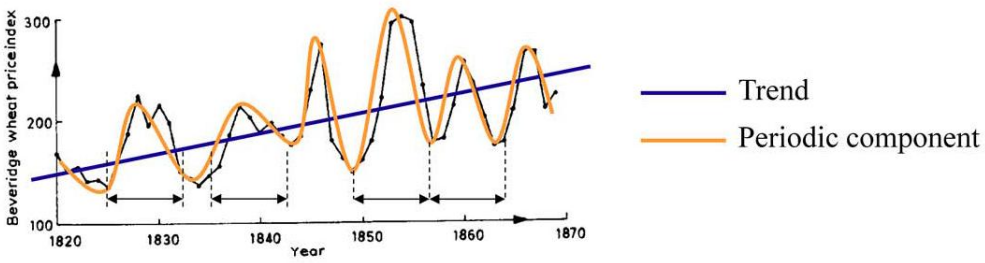


Figure 2.1: The Beveridge wheat price index is the average in nearly 50 places in various countries measured in successive years from 1500 to 1869. [1]

According to Chatfield, most time series exhibit a variation at a fixed period of time (seasonality) such as for example the seasonal variation of temperature. Beyond this cycle, there exists either or both a long term change in the mean (trend) that can be linear, quadratic, and a periodic (cyclic) component. In practice, these 3 features are rarely sufficient for the classification or regression of real time series.

[1] This time series can be downloaded from `http://www.york.ac.uk/depts/maths/data/ts/ts04.dat`

Other authors made the hypothesis of time independency between the observations x_{it}. They consider time series as a static vector data and use classic machine learning algorithms [Lia+12]; [CT01]; [HWZ13]; [HHK12]. Our work focus on classification and regression problems, and on time series comparison through metrics.

2.2 Properties of a metric

A mapping $D : \mathbb{R}^p \times \mathbb{R}^p \rightarrow \mathbb{R}^+$ over a vector space $\mathbb{R}^p$ is called a metric or a distance if for all vectors $\forall \mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_l \in \mathbb{R}^p$, it satisfies the properties:

Comment [CTD8]: je préfère garder l'espace pour + de visibilité

1. $D(\mathbf{x}_i, \mathbf{x}_j) \geq 0$ (positivity)
2. $D(\mathbf{x}_i, \mathbf{x}_j) = D(\mathbf{x}_j, \mathbf{x}_i)$ (symmetry)
3. $D(\mathbf{x}_i, \mathbf{x}_j) = 0 \Leftrightarrow \mathbf{x}_i = \mathbf{x}_j$ (distinguishability)
4. $D(\mathbf{x}_i, \mathbf{x}_j) D(\mathbf{x}_j, \mathbf{x}_l) \leq D(\mathbf{x}_i, \mathbf{x}_l)$ (triangular inequality)

A mapping D that satisfies at least properties 1, 2, 3 is called a dissimilarity, and the one that satisfies at least properties 1, 2, 4 a pseudo-metric. Note that for a metric, a dissimilarity and a pseudo metric, if a time series $\mathbf{x}_i$ is expected to be closer to $\mathbf{x}_j$ than to $\mathbf{x}_l$, then $D(\mathbf{x}_i, \mathbf{x}_j) \leq D(\mathbf{x}_i, \mathbf{x}_l)$. On the contrary, the mapping is called a similarity S when the time series $\mathbf{x}_i$ is expected to be closer to $\mathbf{x}_j$ than to $\mathbf{x}_l$ and then $S(\mathbf{x}_i, \mathbf{x}_j) \geq S(\mathbf{x}_i, \mathbf{x}_l)$. To simplify the discussion in the following, we refer to pseudo-metric and dissimilarity as metrics, pointing out the distinction only when necessary.

2.3 Unimodal metrics for time series

Defining and evaluating metrics for time series has become an active area of research for a wide variety of problems in machine learning [Din+08]; [Naj+12]. In the following, we suppose that time series have the same lengths Q and have been regularly sampled at the frequency f_e. Let $\mathbf{x}_i = (x_{i1}, x_{i2}, ..., x_{iQ})$ and $\mathbf{x}_j = (x_{i1}, x_{i2}, ..., x_{iQ})$ be two univariate time series of length Q.

A large number of distance measures have been proposed in the literature [MV14]. Contrary to static data, time series may exibit modalities and specificities due to their temporal nature (e.g., value, shape, frequency, delay, temporal locality). In this section, we review 3 categories of time series metrics used in our work: amplitude-based, frequential-based and behavior-based.

2.3.1 Amplitude-based metrics

The most usual comparison measures are amplitude-based metrics, where time series are compared in the temporal domain on their amplitudes regardless of their behaviors or frequential characteristics. Among these metrics, there are the commonly used Euclidean distance that compares elements observed at the same time [Din+08]:

$$d_E(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{t=1}^{Q} (x_{it} - x_{jt})^2} \tag{2.1}$$

Note that the Euclidean distance is a particular case of the Minkowski L_p norm ($p = 2$). An other amplitude-based metric is the Mahalanobis distance [PL12], defined as a dissimilarity measure between two random vectors $\mathbf{x}_i$ and $\mathbf{x}_j$ of the same distribution with the covariance matrix $\mathbf{M}$:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M}^{-1} (\mathbf{x}_i - \mathbf{x}_j) \tag{2.2}$$

If the covariance matrix $\mathbf{M}$ is the identity matrix, the Mahalanobis distance is equal to the Euclidean distance. If the covariance matrix $\mathbf{M}$ is diagonal, then the resulting distance measure is called a normalized Euclidean distance:

$$d_M(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{l=1}^{Q} \frac{(x_{il} - x_{jl})^2}{m_l}} \tag{2.3}$$

where m_l is the variance of the x_{il} and x_{jl} over the sample set. In the following of the work, we consider the standard Euclidean distance d_E as the amplitude-based distance d_A.

In the example of Fig. 2.2, the aim is to determined which time series ($\mathbf{x}_2$ or $\mathbf{x}_3$) is the closest to $\mathbf{x}_1$. The amplitude-based distance d_A states that $\mathbf{x}_2$ is closer to $\mathbf{x}_1$ than $\mathbf{x}_3$ since $d_A(\mathbf{x}_1, \mathbf{x}_2) = 7.8816 < d_A(\mathbf{x}_1, \mathbf{x}_3) = 31.2250$.

2.3.2 Frequential-based metrics

The second category, commonly used in signal processing, relies on comparing time series based on their frequential properties (e.g. Fourier Transform, Wavelet, Mel-Frequency Cepstral Coefficients [SS12]; [TC98]; [BM67]). In our work, we limit the frequential comparison to Discrete Fourier Transform [Lhe+11], but other frequential properties can be used as well. Thus, for time series comparison, first the time series $\mathbf{x}_i$ are transformed into their Fourier representation $\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, ..., \tilde{x}_{iF}]$, with $\tilde{x}_{if}$ the complex component at frequential index f. The Euclidean distance is then used between their respective complex number modules $\tilde{x}_{if}$, noted $|\tilde{x}_{if}|$:

$$d_F(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{\sum_{f=1}^{F} (|\tilde{x}_{if}| - |\tilde{x}_{jf}|)^2} \tag{2.4}$$

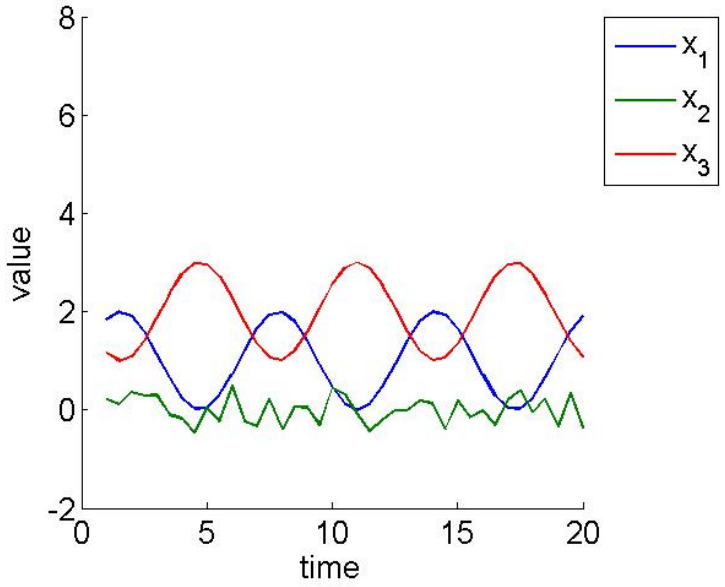
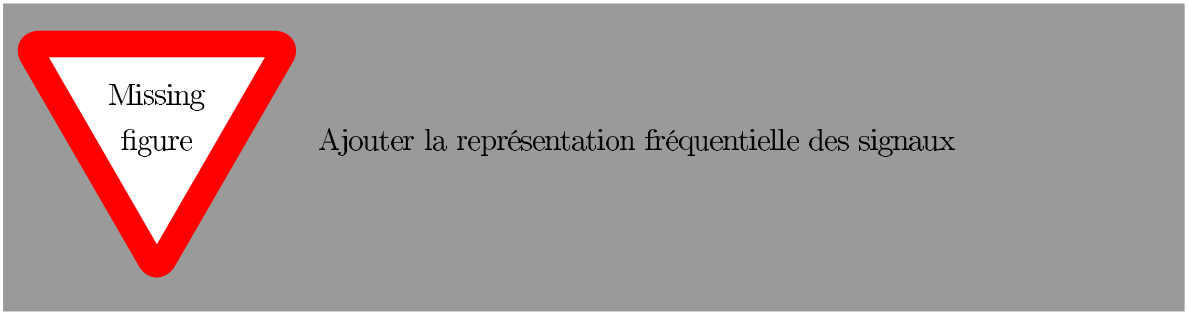


Figure 2.2: 3 toy time series. Time series in blue and red are two sinusoïdal signals. Time series in green is a random signal.

In the example of Fig. 2.2, the frequential-based distance d_F states that the time series $\mathbf{x}_3$ is closer to $\mathbf{x}_1$ than $\mathbf{x}_2$ since $d_F(\mathbf{x}_1, \mathbf{x}_3) = 0.8519 < d_F(\mathbf{x}_1, \mathbf{x}_2) = 0.9250$. This can be illustrated in the Frequency domain (Fig. **??**)



2.3.3 Behavior-based metrics

The third category of metrics aims to compare time series based on their shape or behavior despite the range of their amplitudes. By time series of similar behavior, it is generally intended that for all temporal window $[t, t']$, they increase or decrease simultaneously with the same growth rate. On the contrary, they are said of opposite behavior if for all $[t, t']$, if one time series increases, the other one decreases and (vise-versa) with the same growth rate in absolute value. Finally, time series are considered of different behaviors if they are not similar, nor opposite. Many applications refer to the Pearson correlation [AT10]; [Ben+09] for

behavior comparison. A generalization of the Pearson correlation is introduced in [DCA11]:

$$cort_r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum (x_{it} - x_{it'})(x_{jt} - x_{jt'})}{\sqrt{\sum (x_{it} - x_{it'})^2}\sqrt{\sum (x_{jt} - x_{jt'})^2}} \tag{2.5}$$

where $|t-t'| \leq r$, $r \in [1, ..., T-1]$. The parameter r can be tuned or fixed a priori. It measures the importance of noise in data. For non-noisy data, low orders r is generally sufficient. For noisy data, the practitioner can either use de-noising data technics (Kalman or Wiener filtering [Kal60]; [Wie42]), or fix a high order r.

The temporal correlation *cort* computes the sum of growth rate between $\mathbf{x}_i$ and $\mathbf{x}_j$ between all pairs of values observed at $[t, t']$ for $t' \leq t+r$ (r-order differences). The value $cort_r(\mathbf{x}_i, \mathbf{x}_j) = 1$ means that $\mathbf{x}_i$ and $\mathbf{x}_j$ have similar behavior. The value $cort_r(\mathbf{x}_i, \mathbf{x}_j) = -1$ means that $\mathbf{x}_i$ and $\mathbf{x}_j$ have opposite behavior. Finally, $cort_r(\mathbf{x}_i, \mathbf{x}_j) = 0$ expresses that their growth rates are stochastically linearly independent (different behaviors).

When $r = T - 1$, it leads to the Pearson correlation. As $cort_r$ is a similarity measure, it can be transformed into a dissimilarity measure:

$$d_B(\mathbf{x}_i, \mathbf{x}_j) = \frac{1 - cort_r(\mathbf{x}_i, \mathbf{x}_j)}{2} \tag{2.6}$$

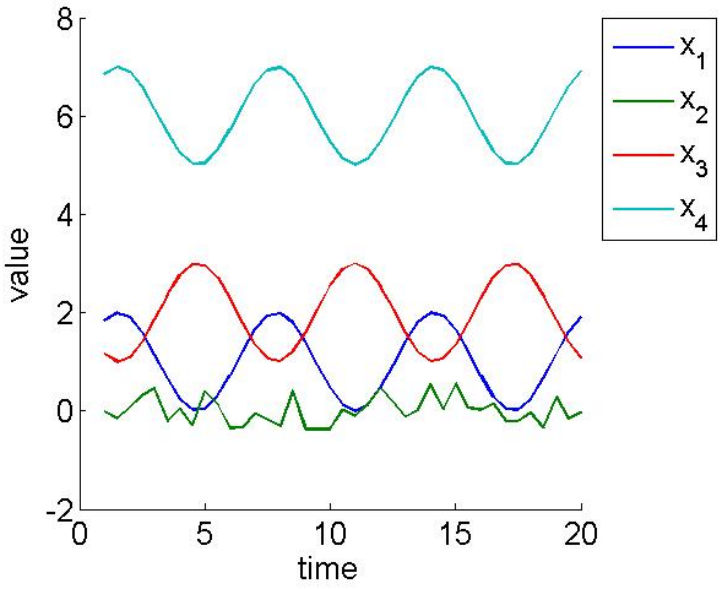


Figure 2.3: The signal from Fig. 2.2 and a signal $\mathbf{x}_4$ which is signal $\mathbf{x}_1$ and an added translation. Based on behavior comparison, $\mathbf{x}_4$ is the closest to $\mathbf{x}_1$.

Now considering Fig. 2.3

$$d_B(\mathbf{x}_1, \mathbf{x}_2) = 0.477$$
$$d_B(\mathbf{x}_1, \mathbf{x}_3) = 1$$
$$d_B(\mathbf{x}_1, \mathbf{x}_4) = 0$$

2.3.4 Other metrics and Kernels for time series

A faire à la fin, pas urgent

- Il existe dans la littérature de nombreuses autres métriques pour les séries temporelles (laisser la porte ouverte).
- Certaines métriques sont utilisées dans le domaine temporelle
- D'autres métriques sont utilisés dans d'autres représentations (Wavelet, etc.)
- Certaines combinent la représentation temporelles et fréquentielles (Représentation spectrogramme en temps-fréquence)
- Se baser sur l'article "TSclust : An R Package for Time Series Clustering".
- Fermer le cadre : dans la suite de notre travail, on ne va pas les utiliser mais elles pourront être intégrées dans le framework qui suivra au chapitre suivant

2.4 Time series alignment and dynamic programming approach

In some applications, time series needs to be compared at different time t (i.e. energy data [Naj+12]) whereas in others, comparing time series on the same time t is essential (i.e. gene expression [DCN07]). When time series are asynchronous (i.e. varying delays or dynamic changes), they must be aligned before any analysis process. The asynchronous effects can be of various natures: time shifting (phase shift in signal processing), time compression or time dilatation. For example, in the case of voice recognition (Fig. 2.4), it is straightforward that a same sentence said by two different speakers will produce different time series: one speaker may speak faster than the other; one speaker may take more time on some vowels, etc.

To cope with delays and dynamic changes, dynamic programming approach has been introduced [BC94]. An alignment $\boldsymbol{\pi}$ of length $|\boldsymbol{\pi}| = m$ between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$ of length T is defined as the set of m ($T \leq m \leq 2T-1$) couples of aligned elements of $\mathbf{x}_i$ to m elements of $\mathbf{x}_j$:

$$\boldsymbol{\pi} = ((\pi_i(1), \pi_j(1)), (\pi_i(2), \pi_j(2)), \ldots, (\pi_i(m), \pi_j(m))) \tag{2.7}$$

Comment [MR9]: Modifier figure. enlever 'one' et mettre la même échelle temporelle

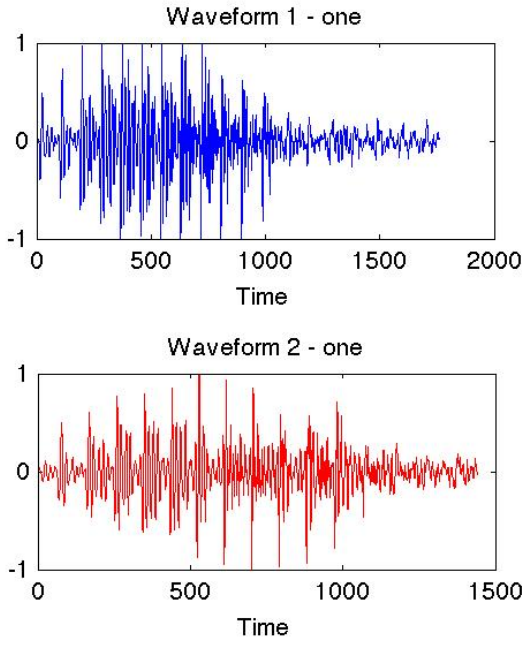


Figure 2.4: Example of a same sentence said by two different speakers. Time series are shifted, compressed and dilatated in the time.

where the applications π_i and π_j defined from $\{1, ..., m\}$ to $\{1, ..., T\}$ obey the following boundary monotonicity conditions:

$$1 = \pi_i(1) \leq \pi_i(2) \leq ... \leq \pi_i(m) = T \tag{2.8}$$
$$1 = \pi_j(1) \leq \pi_j(2) \leq ... \leq \pi_j(m) = T \tag{2.9}$$

$\forall l \in \{1, ..., m\}$,

$$\pi_i(l+1) \leq \pi_i(l) + 1 \tag{2.10}$$
$$\text{and} \quad \pi_j(l+1) \leq \pi_j(l) + 1 \tag{2.11}$$
$$\text{and} \quad (\pi_i(l+1) - \pi_i(l)) - (\pi_j(l+1) - \pi_j(l)) \geq 1. \tag{2.12}$$

Intuitively, an alignment $\boldsymbol{\pi}$ defines a way to associate elements of two time series. Alignments can be described by paths in the $T \times T$ grid that crosses the elements of $\mathbf{x}_i$ and $\mathbf{x}_j$ (Fig. 2.5). We denote $\boldsymbol{\pi}$ a valid alignment and A, the set of all possible alignments between $\mathbf{x}_i$ and $\mathbf{x}_j$ ($\boldsymbol{\pi} \in A$). To find the best alignment $\boldsymbol{\pi}^*$ between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$, the Dynamic Time Warping (DTW) algorithm has been proposed [KR04]; [SC].

DTW requires to choose a cost function φ to be optimised, such as a dissimilarity function (d_A, d_B, d_F, etc.). Classical DTW uses the Euclidean distance d_A (Eq. 2.1) as the cost

function [BC94]. The warp path $\boldsymbol{\pi}$ is optimized for the chosen cost function φ:

$$\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in A}{\operatorname{argmin}} \frac{1}{|\boldsymbol{\pi}|} \sum_{(t,t') \in \boldsymbol{\pi}} \varphi(x_{it}, x_{jt'}) \tag{2.13}$$

When the cost function φ is a similarity measure, the optimization involves maximization instead of minimization. When other constraints are applied on $\boldsymbol{\pi}$, Eq. (2.13) leads to other variants of DTW (Sakoe-Shiba [SC78], Itakura parallelogram [RJ93]). Finally, the warped signals $\mathbf{x}_{i,\boldsymbol{\pi}}$ and $\mathbf{x}_{j,\boldsymbol{\pi}}$ are defined as:

$$\mathbf{x}_{i,\boldsymbol{\pi}} = (x_{i\pi_i(1)}, ..., x_{i\pi_i(m)}) \tag{2.14}$$

$$\mathbf{x}_{j,\boldsymbol{\pi}} = (x_{j\pi_j(1)}, ..., x_{j\pi_j(m)}) \tag{2.15}$$

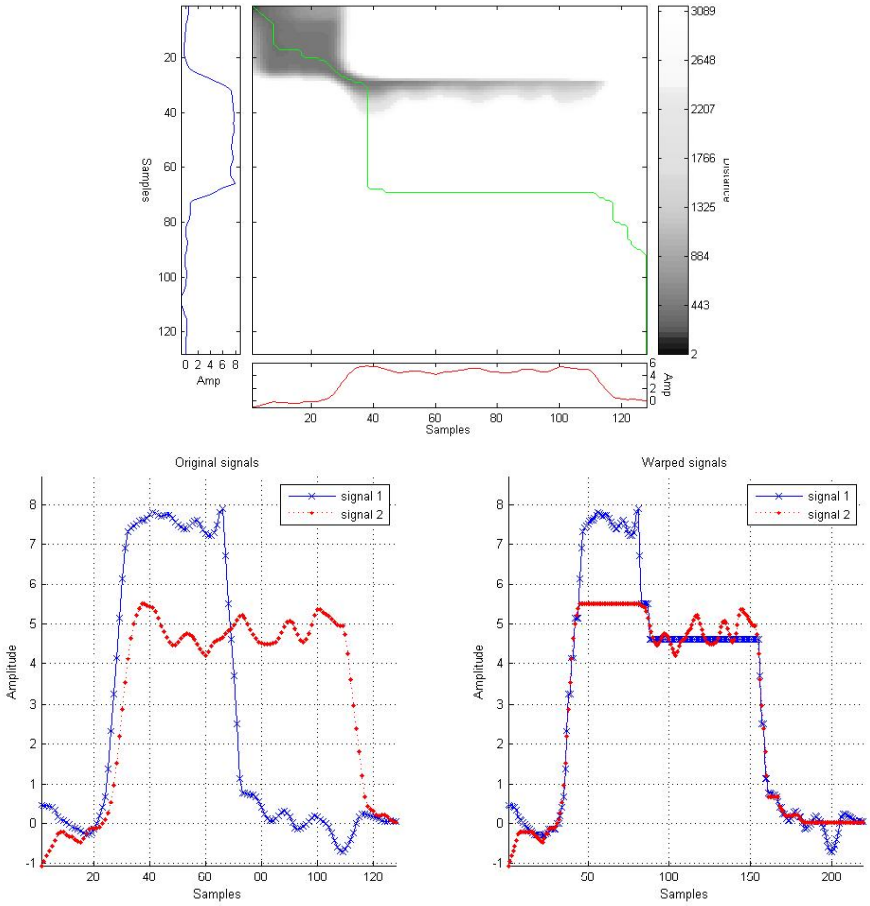


Figure 2.5: Example of DTW grid between 2 time series $\mathbf{x}_i$ and $\mathbf{x}_j$ (top) and the signals before and after warping (bottom). On the DTW grid, the two signals can be represented on the left and bottom of the grid. The optimal path $\boldsymbol{\pi}^*$ is represented in green line and show to associate elements of $\mathbf{x}_i$ to element of $\mathbf{x}_j$. Background show in grey scale the value of the considered metric (amplitude-based distance d_A in classical DTW)

The previous metric (amplitude-based d_A, behavior-based d_B) can be then computed on the warped signals $\mathbf{x}_{i,\boldsymbol{\pi}^*}$ and $\mathbf{x}_{j,\boldsymbol{\pi}^*}$. In the following, we suppose that the best alignment $\boldsymbol{\pi}^*$ is found. For simplification purpose, we refer $\mathbf{x}_{i,\boldsymbol{\pi}^*}$ and $\mathbf{x}_{j,\boldsymbol{\pi}^*}$ as $\mathbf{x}_i$ and $\mathbf{x}_j$.

2.5 Combined metrics for time series

In most classification problems, it is not known a priori if time series of a same class exhibits same characteristics based on their amplitude, behavior or frequential components alone. In some cases, several components (amplitude, behavior and/or frequential) may be implied.

A first technic considers a classifier for each p metric and combines the decision of the p resulting classifiers. This methods is referred as post-fusion , not considered in our work. Other propositions show the benefit of involving both behavior and amplitude components through a combination function. They combines the unimodal metrics together to obtain a single metric used after that in a classifier. This is called pre-fusion. The most classical combination functions combines the unimodal metrics (mainly d_A and d_B) through linear and geometric functions:

$$D_{Lin}(\mathbf{x}_i, \mathbf{x}_j) = \alpha d_B(\mathbf{x}_i, \mathbf{x}_j) + (1-\alpha) d_A(\mathbf{x}_i, \mathbf{x}_j) \tag{2.16}$$

$$D_{Geom}(\mathbf{x}_i, \mathbf{x}_j) = (d_B(\mathbf{x}_i, \mathbf{x}_j))^{\alpha} (d_A(\mathbf{x}_i, \mathbf{x}_j))^{1-\alpha} \tag{2.17}$$

where $\alpha \in [0; 1]$ defines the trade-off between the amplitude d_A and the behavior d_B components, and is thus application dependent. In general, it is learned through a grid search procedure. Without being restrictive, these combinations can be extended to take into account more unimodal metrics.
More specific work on d_A and *cort* propose to combine the two components through a sigmoid combination function [DCA11]:

$$D_{Sig}(\mathbf{x}_i, \mathbf{x}_j) = \frac{2 d_A(\mathbf{x}_i, \mathbf{x}_j)}{1 + \exp(\alpha cort_r(\mathbf{x}_i, \mathbf{x}_j))} \tag{2.18}$$

where α is a parameter that defines the compromise between behavior and amplitude components. When α is fixed to 0, the metric only includes the value proximity component. For $\alpha \geq 6$, the metric completely includes the behavior proximity component.

Fig.2.6 illustrates the value of the resulting combined metrics (D_{Lin}, D_{Geom} and D_{Sig}) in 2-dimensional space using contour plots for different values of the trade-off α. For small value of α ($\alpha = 0$), the three metrics only includes d_A. For high value of α ($\alpha = 1$), D_{Lin} and D_{Geom} only includes d_B. For $\alpha = 6$, D_{Sig} doesn't include completely *cort*. Note that these combinations are fixed and defined independently from the analysis task at hand. Moreover, in the case of D_{Sig}, only two variables are taking into account in these combined metrics and the component $cort_r$ can be seen as a penalizing factor of d_A. It doesn't represent a real compromise between value and behavior components. Finally, by adding metrics, the grid

search to find the best parameters can become time consuming.

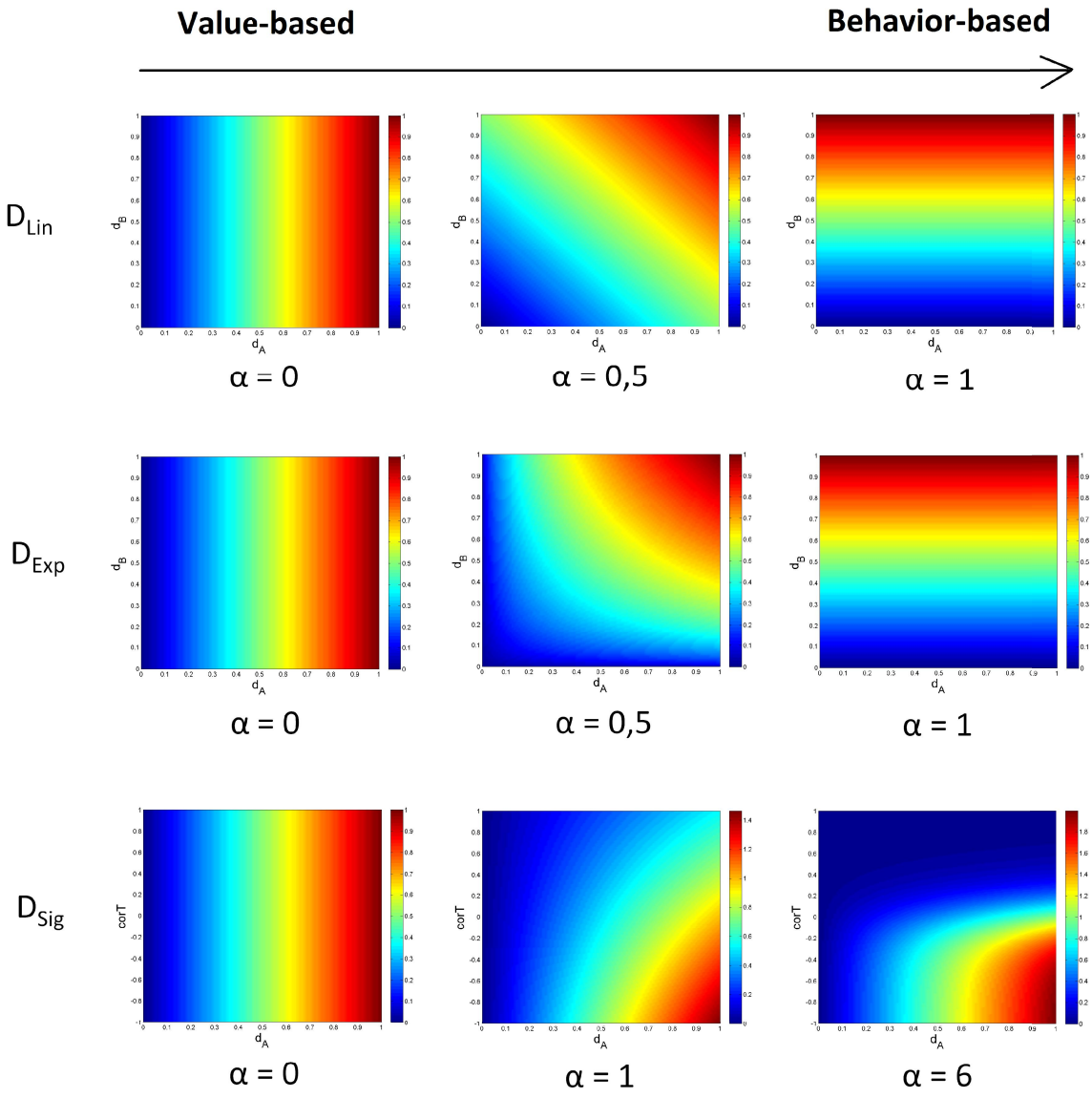


Figure 2.6: Contour plot of the resulting combined metrics: D_{Lin} (1^{st} line), D_{Geom} (2^{nd} line) and D_{Sig} (3^{rd} line), for different value of α (D_{Sig}: $\alpha = 0; 1; 6$ and D_{Lin} and D_{Geom}: $\alpha = 0; 0.5; 1$). For D_{Sig}, the first and second dimensions are respectively the amplitude-based metrics d_A and the temporal correlation $corT$; for D_{Lin} and D_{Geom}, they correspond to d_A and the behavior-based metric d_B.

2.6 Metric learning

As our objective is to learn a metric in order to optimize the performance of the k-NN classifier, we review first metric learning concepts. Then, we focus on the framework proposed by

Weinberger & Saul for Large Margin Nearest Neighbor (LMNN) classification [WS09].

2.6.1 Review on metric learning work

In the case of static data, many work have demonstrated that k-NN classification performances depends highly on the considered metric and can be improved by learning an appropriate metric [She+02]; [Gol+04]; [CHL05]. Metric Learning can be defined as a process that aims to learn a distance from labeled examples by making closer samples that are expected to be similar, and far away those expected to be dissimilar.

A faire, avec papier PRL et papier Aurélien Bellet

2.6.2 Large Margin Nearest Neighbors (LMNN)

Let $\mathbf{X} = \{\mathbf{x}_i, y_i\}_{i=1}^N$ be a set of N static vector samples, $\mathbf{x}_i \in \mathbb{R}^p$, p being the number of descriptive features and y_i the class labels. Weinberger & Saul proposed in [WS09] an approach to learn a dissimilarity metric D for a large margin k-NN in the case of static data.

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

$$D(\mathbf{x}_i, \mathbf{x}_l) \leq D(\mathbf{x}_i - \mathbf{x}_j) + 1 \tag{2.19}$$

Geometrically, an imposter $\mathbf{x}_l$ is a sample that invades the target neighborhood plus one unit margin as illustrated in Fig. 2.7. The target neighborhood is defined with respect to an initial metric. Without prior knowledge, L2-norm is often used. Metric Learning by LMNN aims to minimize the number of impostors invading the target neighborhood. By adding a margin of safety of one, the model is ensured to be robust to small amounts of noise in the training sample (large margin). The learned metric D pulls the targets $\mathbf{x}_j$ and pushes the imposters $\mathbf{x}_l$ as shown in Fig. 2.7.

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

$$D(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{L}(\mathbf{x}_i, \mathbf{x}_j)||_2^2 \tag{2.20}$$

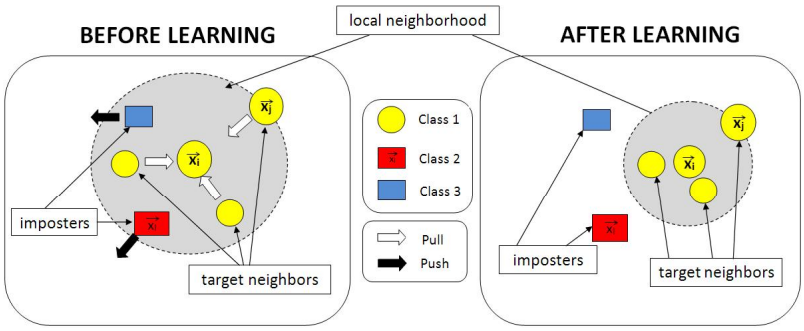


Figure 2.7: Pushed and pulled samples in the $k = 3$ target neighborhood of $\mathbf{x}_i$ before (left) and after (right) learning. The pushed (vs. pulled) samples are indicated by a white (vs. black) arrows (Weinberger & Sault [WS09]).

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

$$\mathbf{M} = \mathbf{L}'\mathbf{L} \tag{2.21}$$

It is proved that any matrix $\mathbf{M}$ formed as below from a real-valued matrix $\mathbf{L}$ is positive semidefinite (i.e., no negative eigenvalues) [WS09]. Using the matrix $\mathbf{M}$, squared distances can be expressed as:

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i - \mathbf{x}_j)\mathbf{M}(\mathbf{x}_i - \mathbf{x}_j) \tag{2.22}$$

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

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = D^2(\mathbf{L}\mathbf{x}_i, \mathbf{L}\mathbf{x}_j) \tag{2.23}$$

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

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

$$\begin{aligned}
&\underset{\mathbf{M},\xi}{\operatorname{argmin}} \underbrace{\sum_{i,j \rightsquigarrow i} D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j)}_{pull} + C \underbrace{\sum_{i,j \rightsquigarrow i, l \twoheadrightarrow i} \frac{1 + y_{il}}{2} . \xi_{ijl}}_{push} \\
&\text{s.t. } \forall j \rightsquigarrow i, l \twoheadrightarrow i, \\
&D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_l) - D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \xi_{ijl} \\
&\xi_{ijl} \geq 0 \\
&\mathbf{M} \succeq 0
\end{aligned} \tag{2.24}$$

where C is a trade-off between the push and pull term and $y_{il} = -1$ if $y_i = y_l$ (same class) and $+1$ otherwise (different classes). Generally, the parameter C is tuned via cross validation and grid search. Similarly to Support Vector Machine (SVM) approach, slack variables ξ_{ijl} are introduced to relax the optimization problem.

2.6.3 Parallels between LMNN and SVM

Many connections can be made between LMNN and SVM: both are convex optimization problem based on a regularized and a loss term. In particular, Do & al. investigate this relationship and have shown that SVM can be formulated as a metric learning problem [Do+12]. The Mahalanobis distance $\mathbf{M}$ learned by LMNN can be expressed as a quadratic mapping ϕ. For a center point $\mathbf{x}_i$, for any sample $\mathbf{x}$, we have [Do+12]:

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}) = D^2(\mathbf{L}\mathbf{x}_i, \mathbf{L}\mathbf{x}) \tag{2.25}$$

$$D^2_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}) = \mathbf{w}_i^T \phi(\mathbf{x}) + b_i \tag{2.26}$$

where $\mathbf{w}_i$ and b_i are the coefficient of the hyperplane H_i in the quadratic space ϕ.

Do & al. show that LMNN can be seen as a set of local SVM classifiers in the quadratic space induced by ϕ. For each center point $\mathbf{x}_i$, LMNN tries in its objective function to have its target neighbors $\mathbf{x}_j$ to have small value $\mathbf{w}_i^T \phi(\mathbf{x}_j) + b_i$, i.e. be at the small distance from the hyperplane H_i. Minimizing the target neighbor distances from the hyperplane H_i makes the distance between support vectors and H_i small. Fig. 2.8 gives the equivalent point of view from the original space (Fig. 2.8(a)) into the quadratic space (Fig. 2.8(b)). The circle $\mathbf{C}_i$ with the center $\mathbf{L}\mathbf{x}_i$ in Fig. 2.8(a) corresponds to the hyperplane H_i in Fig. 2.8(b).

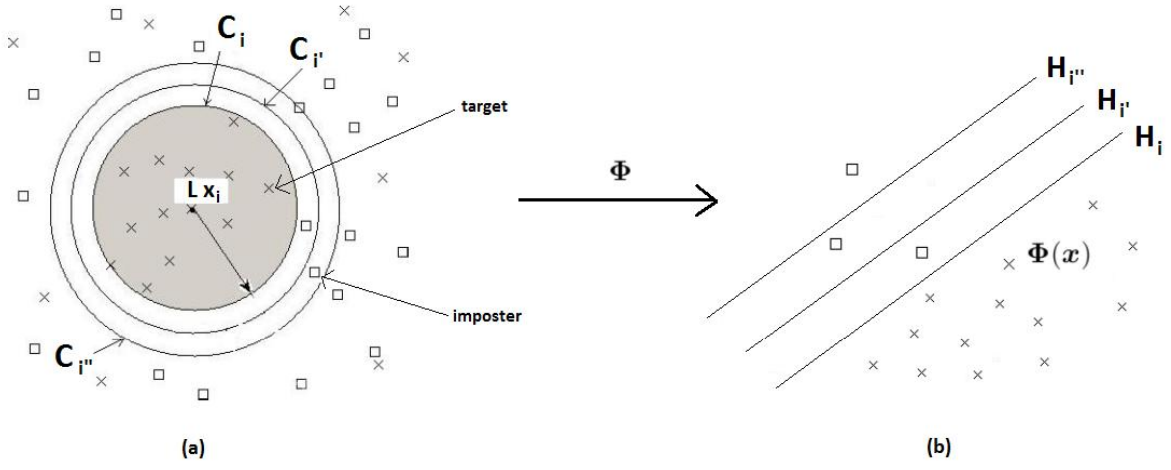


Figure 2.8: (a) Standard LMNN model view (b) LMNN model view under an SVM-like interpretation [Do+12]

Geometrically, SVM margin is defined globally with respect to a hyperplane, while LMNN margin is defined locally with respect to a center point $\mathbf{x}_i$. Fig. 2.9(a) illustrates the different

local linear models in the quadratic space. The optimization process of LMNN combines the different local SVM hyperplane by bringing each point $\phi(\mathbf{x}_i)$ around a consensus hyperplane H.

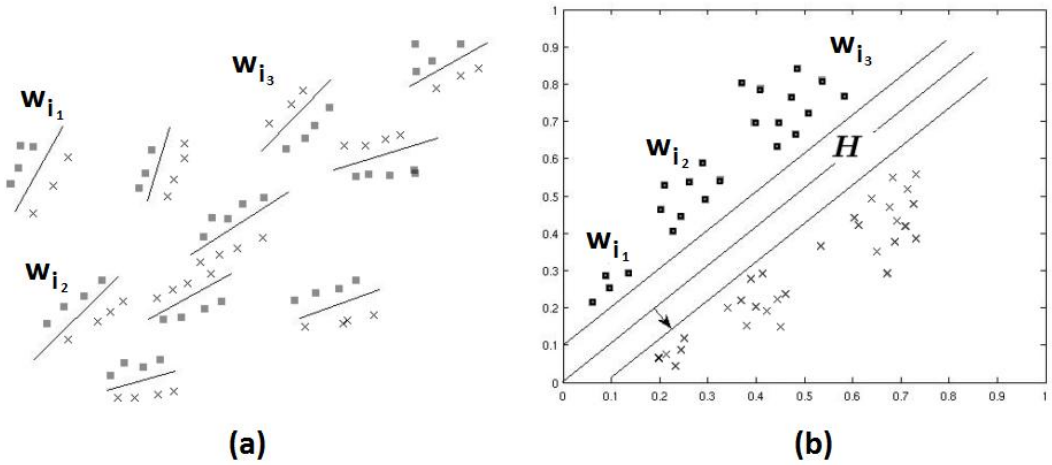


Figure 2.9: (a) LMNN in a local SVM-like view (b) LMNN in an SVM metric learning view [Do+12]

From these connections, some authors extends the LMNN approach to work in non-linear feature spaces by using the "kernel trick" . Finally, note that LMNN differs from SVM in which LMNN requires no modification for multiclass problems. ref

2.7 Conclusion of the chapter

To cope with modalities inherent to time series (amplitude, behavior, frequency, etc.), we review in this chapter several unimodal metrics for time series, in particular, the Euclidean distance d_A, the Temporal correlation d_B or the Fourier-based distance d_F. In practice, real time series may be subjected to delays and need to be re-aligned before any analysis task. For that, the Dynamic Time Warping (DTW) algorithm is used in practice. However, these metrics (d_A, d_B, d_F) only include one modality. In general, several modalities may be implied and authors proposed to combine temporal metrics together. They mainly combine the Euclidean distance d_A and the Temporal correlation d_B.

As k-NN performances is impacted by the choice of the metric, other work propose in the case of static data to learn the metric in order to optimize the k-NN classification. In the following, we extend this framework to learn a combined metric for large margin k-NN classifier of time series.

Conclusion of Part I

In order to make the classification or regression of time series, a lot of technics exist in the literature. Our work focus on the k-NN classifier and the SVM will be used in the following for its large margin concept. We note that the k-Nearest Neighbors algorithm is based on the comparison of time series through distance measures.

Considering time series as static data lead to the only comparison based on their amplitude and the same time. To take into account other specificities of time series (behavior, frequential components), other metrics (e.g., the temporal correlation d_B, the frequential-based distance d_F, etc.) and other methods (Dynamic Time Warping DTW, dichotomy) have been proposed in the literature to cope with temporal characteristics.

Learning an adequate metric is a key challenge to well classify time series. Inspired by Metric Learning work for static data, we propose in the following a framework to learn a Multi-modal and Multi-scale Metric for a robust nearest neighbor classifier of time series.

Part II

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

The first part has enlightened the importance of combining several modalities and several scales to make a better analysis of time series (classification, regression). We propose, in this part, a framework to learn this metric. For that, the idea is to introduce a new space representation, the pairwise space, where a vector is a pair of time series described by several unimodal metrics. Then, we formalize the problem of learning the metric as an optimization problem and show its equivalence by solving an adequate Support Vector Machine (SVM) problem.

In the first chapter, we present this pairwise representation and formalize the optimization problem and its adapted SVM equivalence. In the second chapter, we present the details of the proposed algorithm M^2TML.

CHAPTER 3

Pairwise space and time series metric learning formalization

Sommaire

In this chapter, we formalize the problem of the PhD which is to learn a metric that combines several unimodal metrics for a robust k-NN classifier.
The computation of a metric always implies a pair of samples. We introduce a new space, the pairwise space in which a pair of time series is embedded as a vector described by the different unimodal metrics at different scales. Inspired from the Metric Learning framework, we transpose the Metric Learning problem in the pairwise space to propose a Multi-Modal and Multi-scale Time series Metric Learning (M^2TML) formalism for the classification and regression of time series.

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:

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

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 Figure 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: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]'$. We denote N the number of pairwise vectors $\mathbf{x}_{ij}$ generated by this embedding.

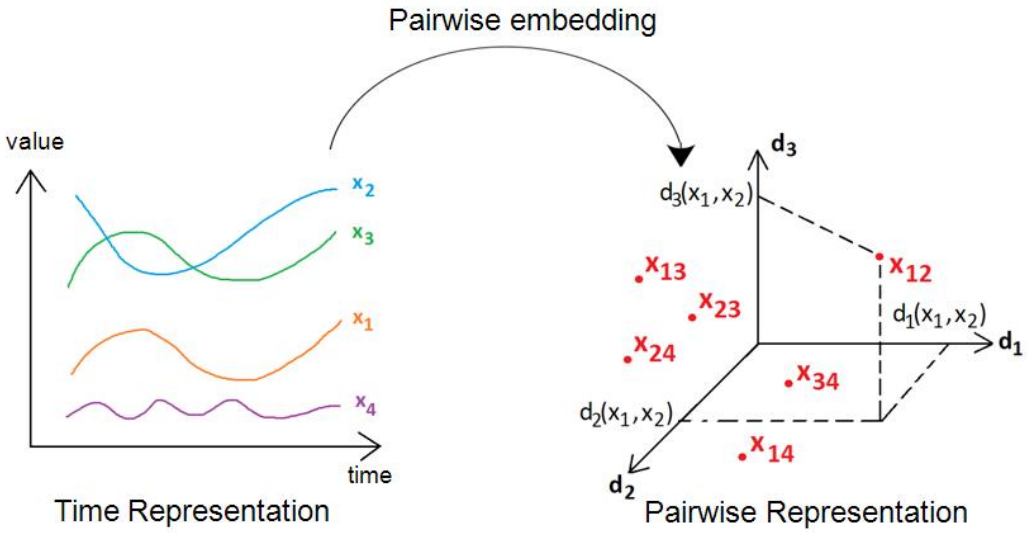


Figure 3.1: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the pairwise space (right). In this example, a pair of time series $(\mathbf{x}_1, \mathbf{x}_2)$ is projected into the pairwise space as a vector $\mathbf{x}_{12}$ described by $p = 3$ basic metrics: $\mathbf{x}_{12}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

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:

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

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

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:

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

For regression problems, each sample $\mathbf{x}_i$ is assigned to a continuous value y_i. Two approaches are possible. The first one aims to discretize by binning 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 practitioner use the classification framework to define the pairwise label y_{ij}.

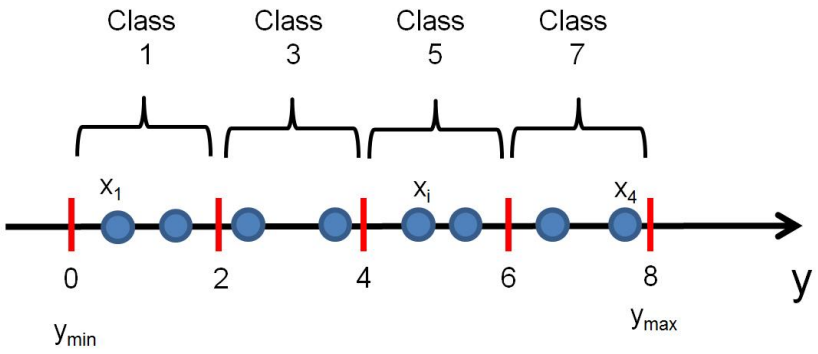
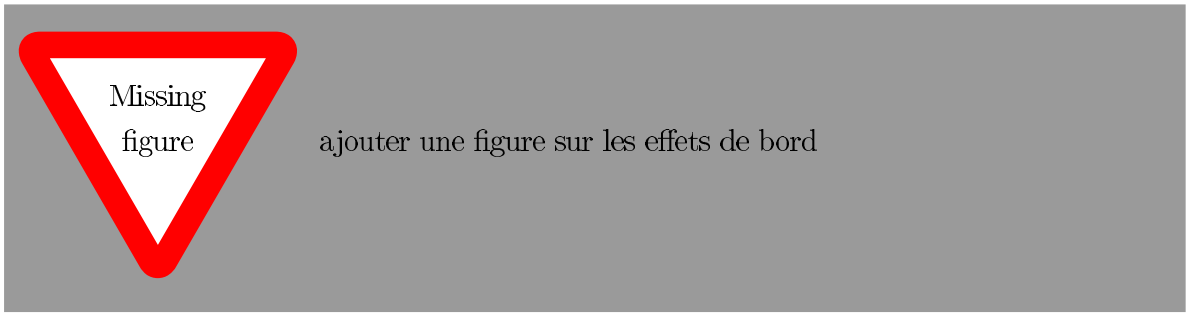


Figure 3.2: Example of discretization by binning a continuous label y into $Q = 4$ equal-length intervals. Each interval is associated to a unique class label. In this example, the class label for each interval is equal to the mean in each interval.

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



The second approach considers the continuous value of y_i, computes a L1-norm between the

labels $|y_i - y_j|$ and compare this value to 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. 3.3):

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

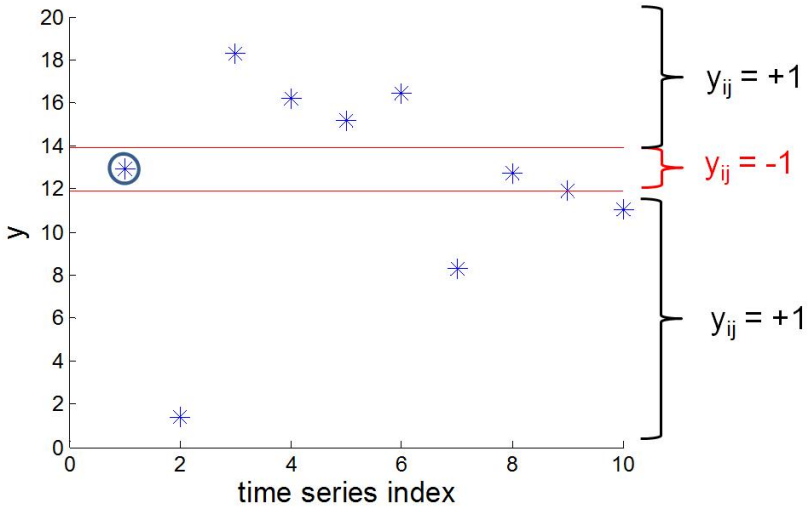


Figure 3.3: Example of pairwise label definition using an ϵ-tube (red lines) around the time series $\mathbf{x}_i$ (circled in blue). For, time series $\mathbf{x}_j$ that falls into the tube, the pairwise label is $y_{ij} = -1$ (similar) and outside of the tube, $y_{ij} = +1$ (not similar).

3.1.3 Interpretation in the pairwise space

When working in the pairwise space, the practitioner has to be careful on the interpretation that can be given in this space because it is not a standard Euclidean space. 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 show an example an two pairwise vectors that are close together in the pairwise space. However, it can be seen in the temporal space that they are not similar at all.

A metric D that combines the p unimodal metric $d_1, \ldots, d_p$ can be seen as a function of this space. Fig. 2.6 has shown the example of the representation of different combined metrics (linear (D_{Lin}), exponential (D_{Exp}) and sigmoid (D_{Sig})) in the pairwise space for two modalities: amplitude-based (d_A) and behavior-based (d_B and $cort$). Finally, it can be noticed that when the time series are embedded in the pairwise, the information of their original class y_i is lost. Any multi-class problem is translated in the pairwise space as a binary classification problem.

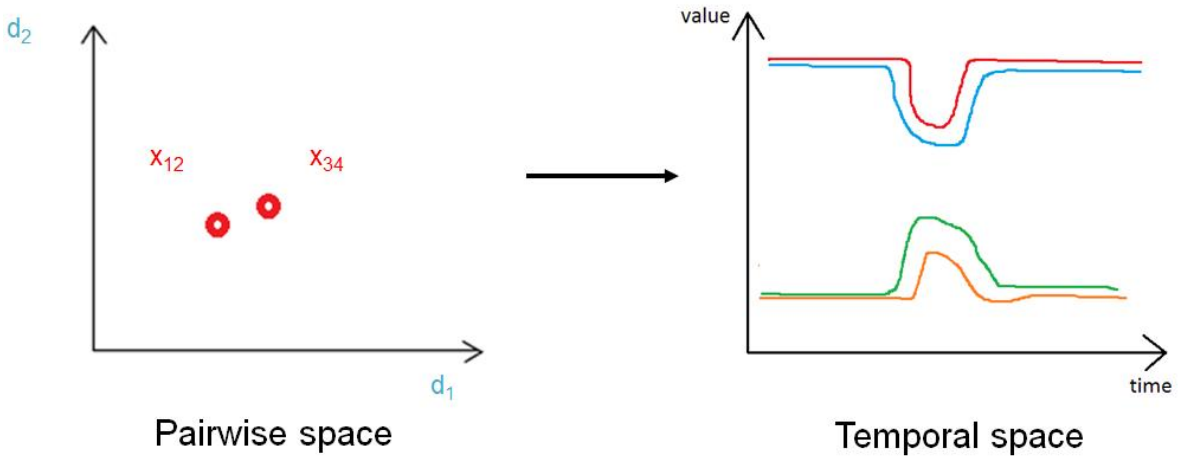


Figure 3.4

3.2 Linear Programming (LP) formalization

Inspired from the Large Margin Nearest Neighbors (LMNN) framework, we transpose the Metric Learning problem into the pairwise space to learn a temporal metric D combining several modalities at different scales. 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 ($y_l \neq y_i$). In the pairwise space, the vector $\mathbf{x}_{ij}$ (targets) are pulled to the origin while the vectors $\mathbf{x}_{il}$ (different classes) are pushed from the origin (Fig. 3.5). The Multi-Modality and Multi-scale Time series Metric Learning (M^2TML) problem is formalized as:

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

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

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

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

where ξ_{ijl} are the slack variables and C, the trade-off between the pull and push costs. M^2TML differs from LMNN in which the push term in M^2TML 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).

By considering a linear combination of d_h: $D(\mathbf{x}_i, \mathbf{x}_j) = \sum_h w_h.d_h(\mathbf{x}_i, \mathbf{x}_j)$, the above formula

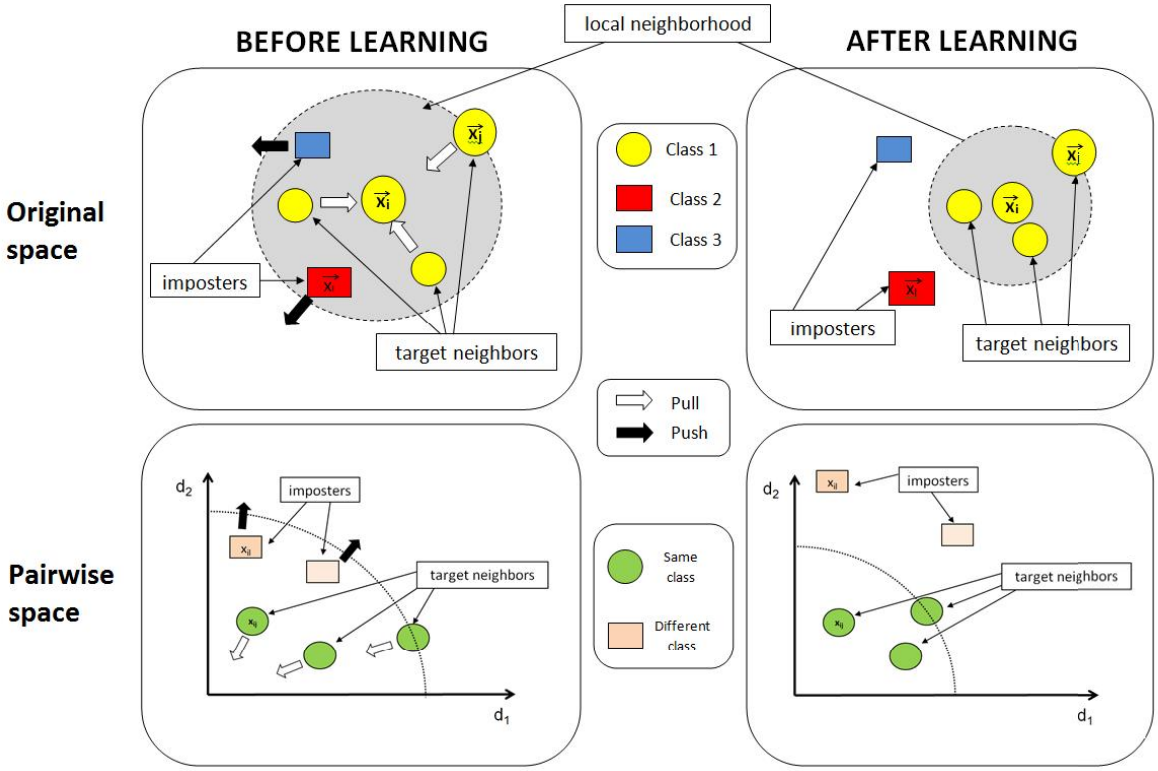


Figure 3.5: Geometric representation of the adaptation of Metric Learning from the original space (top) to the pairwise space (bottom) for for a $k = 3$ target neighborhood of $\mathbf{x}_i$. Before learning (left), imposters $\mathbf{x}_l$ invade the targets perimeter $\mathbf{x}_j$. In the pairwise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

leads to the M^{2}TML primal formulation:

$$\underset{\mathbf{w},\xi}{\operatorname{argmin}} \underbrace{||\mathbf{x}'_{tar}\mathbf{w}||_1}_{pull} + \underbrace{C \sum_{i,j \rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl}}_{pull} \tag{3.8}$$

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

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

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

where $\mathbf{x}_{tar}$ is a $(k.N) \times p$ matrix containing all targets $\mathbf{x}_{ij}$.

M^{2}TML can be seen as a large margin problem in the pairwise space. Many parallels can be done with SVM and the large margin concept. The "pull" term acts as a L1 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 targets $\mathbf{x}_{ij}$ and pairs of different class $\mathbf{x}_{il}$.

To ensure the positivity of the learnt metric D (property 1 in Section 2.2), one possible solution is to set $w_h \geq 0$ for all $h = 1...p$. This constraint can be added into the optimization problem.

3.3 Quadratic Programming (QP) formalization

Similarly to the SVM dual formulation (Section 1.2.2.c), the M^{2}TML primal formulation in Eq. 3.8 can be derived into its dual form to obtain non-linear solutions for D. For that, a change is operated first on the regularization term from a L1 to a L2 norm on $\mathbf{w}$: $\frac{1}{2}||\mathbf{x}'_{tar}\mathbf{w}||_2^2$:

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

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

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

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

This formulation can be reduced to 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 $2 \times N$ constraints (Eqs. 3.12 and 3.13) multiplied by their respective Lagrange multipliers $\boldsymbol{\alpha}$ and $\mathbf{r}$:

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

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:

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

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

that leads to:

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

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

Substituting Eq.3.15 and 3.16 back into $L(\mathbf{w}, \xi, \boldsymbol{\alpha}, \mathbf{r})$, we get the M^{2}TML dual formulation:

$$\underset{\boldsymbol{\alpha}}{\operatorname{argmax}} \sum_{ijl} \alpha_{ijl} - \frac{1}{2} \sum_{ijl} \sum_{i'j'l'} \alpha_{ijl}\alpha_{i'j'l'}(\mathbf{x}_{il} - \mathbf{x}_{ij})'(\mathbf{x}_{tar}\mathbf{x}'_{tar})^{-1}(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'}) \tag{3.17}$$

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

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

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

$$D(\mathbf{x}_{i'j'}) = \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij})'(\mathbf{x}_{tar}\mathbf{x}'_{tar})^{-1}(\mathbf{x}_{i'j'} - \mathbf{x}_{ij}) - \sum_{ijl} \alpha_{ijl}(\mathbf{x}_{il} - \mathbf{x}_{ij})'(\mathbf{x}_{tar}\mathbf{x}'_{tar})^{-1}(\mathbf{0} - \mathbf{x}_{ij}) \tag{3.19}$$

At the optimality, only the triplet $(\mathbf{x}_{il} - \mathbf{x}_{ij})$ 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 points have $\alpha_{ijl} = 0$ (non-support vector), and the function 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:

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

from which we deduce that either $\mathbf{w}'(\mathbf{x}_{il} - \mathbf{x}_{ij}) > 1$ and $\alpha_{ijl} = 0$ (the triplet $(\mathbf{x}_{il} - \mathbf{x}_{ij})$ is a non-support vector), or $\mathbf{w}'(\mathbf{x}_{il} - \mathbf{x}_{ij}) = 1 - \xi_{ijl}$ and $\alpha_{ijl} > 0$ (the triplet is a support vector). Therefore, 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.

Note that the dual formulation in Eq. 3.17 only relies on the inner product $(\mathbf{x}_{i'l'} - \mathbf{x}_{i'j'})'(\mathbf{x}_{tar}\mathbf{x}'_{tar})^{-1}(\mathbf{x}_{il} - \mathbf{x}_{ij})$. We can hence apply the kernel trick to find non-linear solutions 3.21:

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

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

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 Motivation

Many parallels have been made between Large Margin Nearest Neighbors (LMNN) and SVM approaches . Similarly, M^{2}TML can be linked to SVM: both are convex optimization problem ref based on a regularized and a loss term. SVM have been well implemented and well studied in the literature for its generalization properties and extension to non-linear solutions. SVM framework is thus well-known. Motivated by these advantages, we propose to solve the M^{2}TML problem by solving an adequate SVM problem. Using this approach, we can naturally extend M^{2}TML problem to find non-linear solutions for the metric D thanks to the 'kernel trick'. In the next section, we demonstrate the equivalence between LP/QP and SVM formulation.

3.4.2 Equivalence between LP/QP and SVM formulation

For a sample $\mathbf{x}_i$, we define the set $\mathbf{X}_{pi} = \{(\mathbf{x}_{ij}, y_{ij}) \text{ s.t. } j \rightsquigarrow i \text{ or } y_{ij} = +1\}$. It corresponds for a sample $\mathbf{x}_i$ to the set of target samples $\mathbf{x}_j$ 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 $\mathbf{X}_p$ [1]:

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

where p_i^- is the half of the number of targets of $\mathbf{x}_i$ and p_i^+ is the half of the number of time series of a different class than $\mathbf{x}_i$:

$$p_i^- = \frac{k}{2} \qquad p_i^+ = \frac{1}{2} \sum_l \frac{1 + y_{il}}{2}$$

We show in the following that solving the SVM problem in Eq. 3.22 for $\mathbf{w}$ and b solves the similar M^{2}TML problem in Eq. 3.11 for $D(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(\mathbf{w}.\mathbf{x}_{ij} + b)$.

First, we recall the constraints in Eq. 3.22:

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

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

These constraints can be split into two sets of constraints:

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

which is equivalent to:

$$\begin{aligned} -(\mathbf{w}.\mathbf{x}_{ij} + b) &\geq 1 - \xi_{ij} \\ (\mathbf{w}.\mathbf{x}_{il} + b) &\geq 1 - \xi_{il} \end{aligned}$$

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

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

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

$$\begin{cases} \bullet \forall i, j, k, l \text{ such that } y_{ij} = -1, \text{ and } y_{kl} = +1, i \neq j \text{ and } i \neq k: \\ D(\mathbf{x}_k, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{kl} + \xi_{ij}}{2} \\ \bullet \forall i, j, l \text{ such that } y_{ij} = -1, \text{ and } y_{il} = +1, i \neq j: \\ D(\mathbf{x}_i, \mathbf{x}_l) - D(\mathbf{x}_i, \mathbf{x}_j) \geq 1 - \frac{\xi_{il} + \xi_{ij}}{2} \end{cases} \tag{3.23}$$

By defining $\xi_{ijl} = \frac{\xi_{ij} + \xi_{il}}{2}$, the second constraint in Eq. 3.23 is the same as the constraints in Eq. 3.12.

It can also be noted that:

$$\bullet \sum_{i,j,y_{ij}=+1} p_i^+ \xi_{ij} = \sum_{il} p_i^+ \frac{1 + y_{il}}{2} \xi_{il} = \frac{1}{2} \sum_{i,j \rightsquigarrow i,l} \frac{1 + y_{il}}{2} \xi_{il}$$

$$\bullet \sum_{i,j,y_{ij}=-1} p_i^- \xi_{ij} = \sum_{i,j \rightsquigarrow i} p_i^- \xi_{ij} = \frac{1}{2} \sum_{i,j \rightsquigarrow i,l} \frac{1 + y_{il}}{2} \xi_{ij}$$

The objective function becomes:

$$\min_{\mathbf{w},\xi} \frac{1}{2}\mathbf{w}'\mathbf{w} + C \sum_{i,j \rightsquigarrow i,l} \frac{1 + y_{il}}{2} \frac{\xi_{ij} + \xi_{il}}{2}$$

The loss-function part of the SVM problem is similar to the one in Eq. 3.11. We can therefore use such SVMs with kernels to find non-linear forms for D:

$$D(\mathbf{x}_{i'}, \mathbf{x}_{j'}) = \frac{1}{2} \left(\sum_{ij} \alpha_{ij} y_{ij} K\left(\mathbf{x}_{ij}, \mathbf{x}_{i'j'}\right) + b \right) \tag{3.24}$$

3.4.3 Relationships between LP/QP and SVM problems.

From last section, it can be noted that the two problems have similarities but also exhibits differences.

First, 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. 3.22), the regularization part tends to minimize the norm of $\mathbf{w}$ whereas in M^{2}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.

Second, an additional set of constraints is present in the SVM formulation (first set of constraints in Eq. 3.23) and not in M^{2}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). This is more constraining for the resulting metric D especially if the neighborhoods have different shapes or are spread unevenly. To overcome this issue, we propose to scale all target spheres to 1 in the preprocessing, such that the risk of over-constraining the problem is very much mitigated.

3.4.4 Geometric interpretation

We show below the QP and SVM resolutions of a 2-NN problem with 2 neighborhoods. For QP, the problem is first solved for each neighborhood independently (blue and red) and then globally (gray). Support vectors of the global problem considering all triplets are indicated with arrows. In general, the global QP solution is different from the best local solution. Also, we notice that the solution found with a Gaussian kernel goes against the monotonicity property - hence bringing closer points that are very far away in the pairwise space.

3.5 Conclusion of the chapter

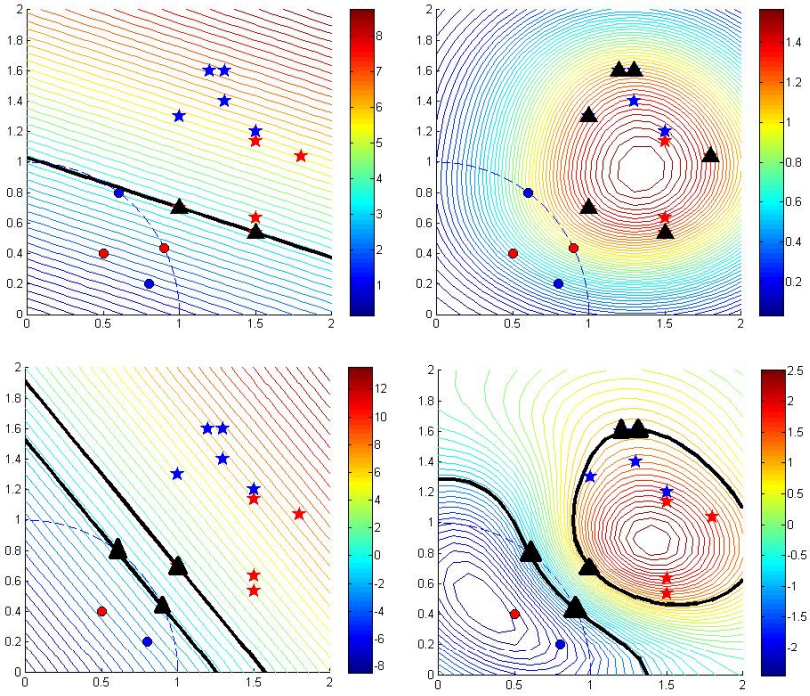


Figure 3.6: Solutions found by solving the QP problem (1st line) and the SVM problem (2nd line), using a linear (1st column) and a Gaussian (2nd column). For the QP resolution, red and blue lines shows the margin when solving the problem for each neighborhood (red and blue points) separately. The global margin is indicated in gray and the metric is represented in color levels. For the SVM, the black lines indicates the SVM margin.

CHAPTER 4

M^{2}TML implementation

Sommaire

Chapeau introductif :

- Quel problème on résout?
- Donner les étapes principales de résolution (sous forme de puces). Cela doit rester général, clair et concis.
- Développer dans chaque section les puces énumérés précédemment.

4.1 Multi-scale comparison

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 (UMD dataset) in which the time series of different classes seems to be similar on a global scale. However, at a more locally scale, a characteristic bell (up or down) at the beginning or at the end of the time series allows to differentiate the classes. 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 makes the metric computation faster.

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 ref

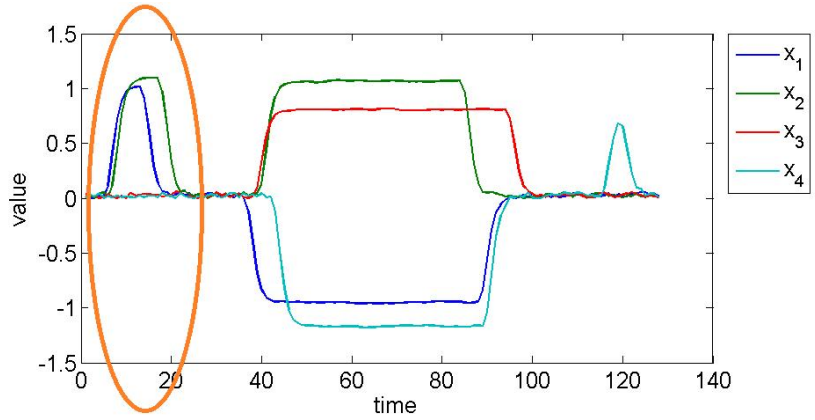


Figure 4.1: Example of 4 time series from the BME dataset, made of 3 classes : Begin, Middle and End. The 'Up' class has a characteristic bell at the beginning of the time series. The 'End' class has a characteristic bell at the end of the time series. The 'Middle' class has no characteristic bell. Orange 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.

methods have been proposed covering the extraction of local features from temporal windows [BC94] or the matching of queries according to a reference sequence [FRM94]. Our work will focus on the computation of local metrics.

It can be noted that the distance measures (d_A[1], d_F, 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. 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 is 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 fix the number of segments and consider a binary segmentation. Let $I = [a; b]$ be a temporal interval of size $(b - a)$. For a strict division (no overlapping), the dichotomy process divide I into two equal intervals at $\frac{b-a}{2}$: the left one I_L and the right I_R one. We add a parameter α that allows to overlap the two intervals I_L and I_R, covering discriminating subsequences in the central region of I (around $\frac{b-a}{2}$) and thus avoiding 'border effects':

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

For $\alpha = 0.6$, the overlap covers 10% of the size of the interval I. A multi-scale description is then obtained on computing the usual time series metrics (d_A, d_B, d_F) on the resulting segments I, I_L and I_R and by repeating the process on I_L and I_R. For a multi-scale

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

amplitude-based comparison based on binary segmentation, Figure 4.2 shows the set of involved amplitude-based measures d_A^{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.

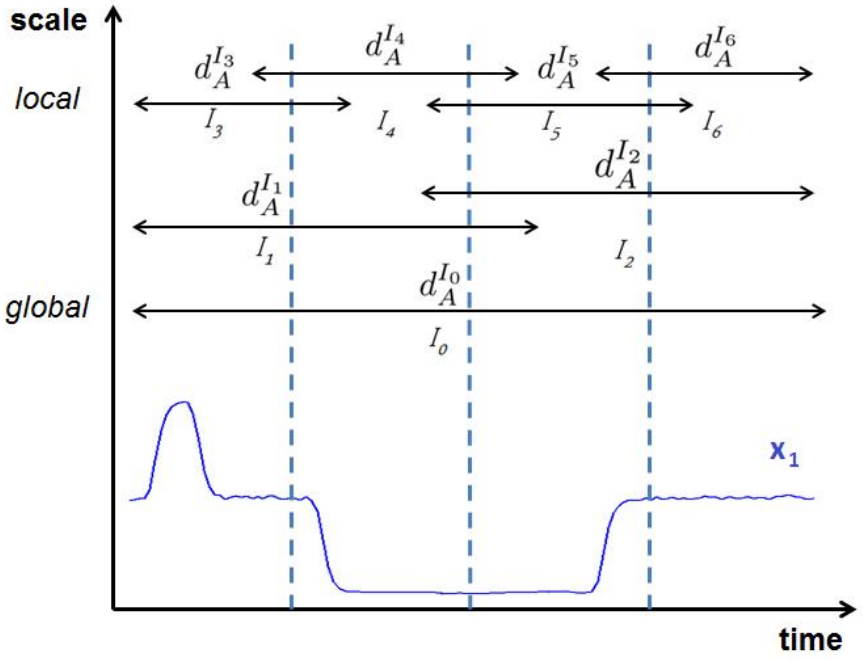


Figure 4.2: Multi-scale amplitude-based measures d_A^{Is}

4.2 Projection in the pairwise space

- Projection
- Log normalization

Pairwise space normalization
This operation is performed 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.

4.3 M-NN M-diff strategy

- Expliquer les différentes stratégies (k-NN VS All / M-NN VS M-diff / k-NN VS Imposters)
- Expliquer pourquoi on va choisir une stratégie M-NN VS M-diff

4.4 Radius normalization

- Expliquer le problème de la non-homogénéité des radius.
- Expliquer comment on résout ce problème par une normalisation des radius de chaque voisinage.

4.5 Solving the SVM problem

- Expliquer l'apprentissage avec le SVM.
- Utilisation de la version L1 du SVM pour avoir une solution sparse.

4.6 Definition of the dissimilarity measure

- Produit scalaire
- Papier PR : norme pondérée x fonction exponentielle
- Version Sylvain : norme x fonction exponentielle?

4.7 Extension to regression problem

(To do)

4.8 Extension to multivariate problem

(To do)

Conclusion of Part II

Part III

Experiments

CHAPTER 5

Experiments

Sommaire

Chapeau introductif

- Application sur des bases de séries temporelles univariés de la littérature (Keogh)
- Données Schneider? ou Expliquer les problématiques de Schneider

5.1 Dataset presentation

5.2 Experimental protocol

5.3 Results

5.4 Discussion

Conclusion of Part III

Conclusion and perspectives

- Bilan des apports de la thèse
- Perspectives
  - Multi-pass learning
  - Kernel pour la résolution du problème QP
  - Utilisation de la distance apprise dans d'autres algorithmes de machine learning (Arbre de décision) pour obtenir une interprétabilité?
  - Utilisation d'autres distances (wavelets, etc.)
  - Apprentissage locale de la métrique

Appendix A

Detailed presentation of the datasets

Appendix B

Solver library

Appendix C

SVM library

Bibliography

[ABR64] M. Aizerman, E. Braverman, and L. Rozonoer. "Theoretical foundations of the potential function method in pattern recognition learning." In: *Automation and Remote Control* 25 (1964), pp. 821–837 (cit. on p. 21).

[Alt92] Ns Altman. "An introduction to kernel and nearest-neighbor nonparametric regression." In: *The American Statistician* 46.3 (1992), pp. 175–185 (cit. on p. 16).

[AT10] Z. Abraham and P.N. Tan. "An Integrated Framework for Simultaneous Classification and Regression of Time-Series Data." In: *ACM SIGKDD*. 2010 (cit. on p. 35).

[BC94] Donald Berndt and James Clifford. "Using dynamic time warping to find patterns in time series." In: *Workshop on Knowledge Knowledge Discovery in Databases* 398 (1994), pp. 359–370 (cit. on pp. 37, 39, 64).

[Ben+09] J. Benesty et al. "Pearson correlation coefficient." In: *Noise Reduction in Speech Processing* (2009) (cit. on p. 35).

[BGV92] Bernhard E. Boser, Isabelle M. Guyon, and Vladimir N. Vapnik. "A Training Algorithm for Optimal Margin Classifiers." In: *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*. 1992, pp. 144–152 (cit. on pp. 17, 22).

[Bis06] Christopher M Bishop. *Pattern Recognition and Machine Learning*. Vol. 4. 4. 2006, p. 738. arXiv: 0-387-31073-8 (cit. on pp. 8, 28).

[BM67] E. O. Brigham and R. E. Morrow. "The fast Fourier transform." In: *Spectrum, IEEE* 4.12 (1967), pp. 63 –70 (cit. on p. 34).

[BMP02] Serge Belongie, Jitendra Malik, and Jan Puzicha. "Shape Matching and Object Recognition Using Shape Contexts." In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (2002), pp. 509–522 (cit. on p. 16).

[CH67] T. Cover and P. Hart. "Nearest neighbor pattern classification." In: *IEEE Transactions on Information Theory* 13.1 (1967), pp. 21–27 (cit. on p. 15).

[Cha04] Christopher Chatfield. *The analysis of time series : an introduction*. 2004, xiii, 333 p. (Cit. on p. 32).

[CHL05] Sumit Chopra, Raia Hadsell, and Yann LeCun. "Learning a similarity metric discriminatively, with application to face verification." In: *CVPR*. Vol. 1. 2005, pp. 539–546 (cit. on p. 42).

[CHY96] Ming Syan Chen, Jiawei Han, and Philip S. Yu. *Data mining: An Overview from a Database Perspective*. 1996 (cit. on p. 8).

[Coc77] William C Cochran. "Snedecor G W & Cochran W G. Statistical methods applied to experiments in agriculture and biology. 5th ed. Ames, Iowa: Iowa State University Press, 1956." In: *Citation Classics* 19 (1977), p. 1 (cit. on p. 12).

[CS01] Koby Crammer and Yoram Singer. "On the Algorithmic Implementation of Multiclass Kernel-based Vector Machines." In: *Journal of Machine Learning Research* 2 (2001), pp. 265–292 (cit. on p. 27).

[CT01] Lijuan Cao and Francis E H Tay. "Financial Forecasting Using Support Vector Machines." In: *Neural Computing & Applications* (2001), pp. 184–192 (cit. on p. 33).

[CV95] Corinna Cortes and Vladimir Vapnik. "Support-vector networks." In: *Machine Learning* 20.3 (1995), pp. 273–297. arXiv: `arXiv:1011.1669v3` (cit. on p. 17).

[CY11] Colin Campbell and Yiming Ying. *Learning with Support Vector Machines*. Vol. 5. 1. 2011, pp. 1–95 (cit. on pp. 17, 21).

[DCA11] A. Douzal-Chouakria and C. Amblard. "Classification trees for time series." In: *Pattern Recognition journal* (2011) (cit. on pp. 36, 40).

[DCN07] A. Douzal-Chouakria and P. Nagabhushan. "Adaptive dissimilarity index for measuring time series proximity." In: *Advances in Data Analysis and Classification* (2007) (cit. on p. 37).

[Den95] T. Denoeux. "A k-nearest neighbor classification rule based on Dempster-Shafer theory." In: *IEEE Transactions on Systems, Man, and Cybernetics* 25.5 (1995), pp. 804–813 (cit. on p. 16).

[DHB95] Thomas G. Dietterich, Hermann Hild, and Ghulum Bakiri. "A comparison of ID3 and backpropagation for English text-to-speech mapping." In: *Machine Learning* 18.1 (1995), pp. 51–80 (cit. on p. 12).

[Die97] T. Dietterich. "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms." In: (1997) (cit. on p. 12).

[Din+08] Hui Ding et al. "Querying and Mining of Time Series Data : Experimental Comparison of Representations and Distance Measures." In: *Proceedings of the VLDB Endowment* 1.2 (2008), pp. 1542–1552. arXiv: `1012.2789v1` (cit. on pp. 16, 33, 34).

[Do+12] Huyen Do et al. "A metric learning perspective of SVM: on the relation of LMNN and SVM." In: *Proceedings of the 15th International Con- ference on Artificial Intelligence and Statistics (AISTAS '12)* (2012), pp. 308–317. arXiv: `arXiv:1201.4714v1` (cit. on pp. 44, 45).

[Dud76] Sahibsingh a. Dudani. "DISTANCE-WEIGHTED k-NEAREST-NEIGHBOR RULE." In: *IEEE Transactions on Systems, Man and Cybernetics* SMC-6.4 (1976), pp. 325–327 (cit. on p. 16).

[FCH08] RE Fan, KW Chang, and CJ Hsieh. "LIBLINEAR: A library for large linear classification." In: *The Journal of Machine Learning* (2008) (cit. on p. 23).

[FRM94] Christos Faloutsos, M. Ranganathan, and Yannis Manolopoulos. "Fast subsequence matching in time-series databases." In: *ACM SIGMOD Record* 23.2 (1994), pp. 419–429 (cit. on p. 64).

[Gol+04] Jacob Goldberger et al. "Neighbourhood Components Analysis." In: *Advances in Neural Information Processing Systems* (2004), pp. 513–520 (cit. on p. 42).

[HCL08] Chih-Wei Hsu, Chih-Chung Chang, and Chih-Jen Lin. "A Practical Guide to Support Vector Classification." In: *BJU international* 101.1 (2008), pp. 1396–400. arXiv: `0-387-31073-8` (cit. on p. 14).

[HHK12] Seok Hwan Hwang, Dae Heon Ham, and Joong Hoon Kim. "Forecasting performance of LS-SVM for nonlinear hydrological time series." In: *KSCE Journal of Civil Engineering* 16.5 (2012), pp. 870–882 (cit. on p. 33).

[HHP01] B Heisele, P Ho, and T Poggio. "Face recognition with support vector machines: global versus component-based approach." In: *IEEE International Conference on Computer Vision, ICCV*. Vol. 2. July. 2001, pp. 688–694 (cit. on p. 17).

[HWZ13] Jianming Hu, Jianzhou Wang, and Guowei Zeng. "A hybrid forecasting approach applied to wind speed time series." In: *Renewable Energy* 60 (2013), pp. 185–194 (cit. on p. 33).

[JMF99] a. K. Jain, M. N. Murty, and P. J. Flynn. "Data clustering: a review." In: *ACM Computing Surveys* 31.3 (1999), pp. 264–323. arXiv: `arXiv:1101.1881v2` (cit. on p. 8).

[Kal60] R E Kalman. "A New Approach to Linear Filtering and Prediction Problems." In: *Transactions of the ASME Journal of Basic Engineering* 82.Series D (1960), pp. 35–45 (cit. on p. 36).

[KGG85] James M. Keller, Michael R. Gray, and James a. Givens. *A fuzzy K-nearest neighbor algorithm*. 1985 (cit. on p. 16).

[KR04] Eamonn Keogh and Chotirat Ann Ratanamahatana. "Exact indexing of dynamic time warping." In: *Knowledge and Information Systems* 7.3 (2004), pp. 358–386 (cit. on p. 38).

[KU02] B Kijsirikul and N Ussivakul. "Multiclass Support Vector Machines using Adaptive Directed Acyclic Graph." In: *Neural Networks, 2002. IJCNN '02. Proceedings of the 2002 International Joint Conference on* 1 (2002), pp. 980–985 (cit. on p. 27).

[Lhe+11] S. Lhermitte et al. "A comparison of time series similarity measures for classification and change detection of ecosystem dynamics." In: *Remote Sensing of Environment* 115.12 (2011), pp. 3129–3152 (cit. on p. 34).

[Lia+12] Chunquan Liang et al. "Learning very fast decision tree from uncertain data streams with positive and unlabeled samples." In: *Information Sciences* 213 (2012), pp. 50–67 (cit. on p. 33).

[MV14] Pablo Montero and José Vilar. "TSclust : An R Package for Time Series Clustering." In: *Journal of Statistical Software November* 62.1 (2014) (cit. on p. 33).

[Naj+12] H. Najmeddine et al. "Mesures de similarité pour l'aide à l'analyse des données énergétiques de bâtiments." In: *RFIA*. 2012 (cit. on pp. 31, 33, 37).

[Ngu+12] L. Nguyen et al. "Predicting collective sentiment dynamics from time-series social media." In: *WISDOM*. 2012 (cit. on p. 31).

[OE73] Richard O Duda and Peter E Hart. *Pattern Classification and Scene Analysis*. Vol. 7. 1973, p. 482 (cit. on pp. 8, 11, 16).

[PAN+08] COSTAS PANAGIOTAKIS et al. "SHAPE-BASED INDIVIDUAL/GROUP DETECTION FOR SPORT VIDEOS CATEGORIZATION." In: *International Journal of Pattern Recognition and Artificial Intelligence* 22.06 (2008), pp. 1187–1213 (cit. on p. 31).

[PL12] Zoltán Prekopcsák and Daniel Lemire. "Time series classification by class-specific Mahalanobis distance measures." In: *Advances in Data Analysis and Classification* 6.3 (2012), pp. 185–200. arXiv: `1010.1526` (cit. on p. 34).

[Ram+08] E. Ramasso et al. "Human action recognition in videos based on the transferable belief model : AAAApplication to athletics jumps." In: *Pattern Analysis and Applications* 11.1 (2008), pp. 1–19 (cit. on p. 31).

[RJ93] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition*. Vol. 103. 1993 (cit. on p. 39).

[SC] Stan Salvador and Philip Chan. "FastDTW : Toward Accurate Dynamic Time Warping in Linear Time and Space." In: () (cit. on p. 38).

[SC78] H. Sakoe and S. Chiba. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition." In: *IEEE transactions on acoustics, speech, and signal processing* (1978) (cit. on p. 39).

[She+02] Noam Shental et al. "Adjustment Learning and Relevant Component Analysis." In: *European Conference on Computer Vision (ECCV)* 2353 (2002), pp. 776–790 (cit. on p. 42).

[SJ89] B W Silverman and M C Jones. "E. Fix and J.L. Hodges (1951): An Important Contribution to Nonparametric Discriminant Analysis and Density Estimation: Commentary on Fix and Hodges (1951)." In: *International Statistical Review / Revue Internationale de Statistique* 57.3 (1989), pp. 233–238 (cit. on p. 15).

[SS12] Md Sahidullah and Goutam Saha. "Design, analysis and experimental evaluation of block based transformation in MFCC computation for speaker recognition." In: *Speech Communication* 54.4 (2012), pp. 543–565 (cit. on p. 34).

[SS13] Bernhard Schlkopf and Alexander J. Smola. *Learning with Kernels*. Vol. 53. 2013, pp. 1689–1699. arXiv: `arXiv:1011.1669v3` (cit. on pp. 17, 23).

[SSB03] Javad Sadri, Ching Y Suen, and Tien D. Bui. "Application of Support Vector Machines for recognition of handwritten Arabic/Persian digits." In: *Second Conference on Machine Vision and Image Processing & Applications (MVIP 2003)* 1 (2003), pp. 300–307 (cit. on p. 17).

[TC98] Christopher Torrence and Gilbert P. Compo. "A Practical Guide to Wavelet Analysis." In: *Bulletin of the American Meteorological Society* 79.1 (1998), pp. 61–78 (cit. on p. 34).

[Wan02] Jung-Ying Wang. "Support Vector Machines (SVM) in bioinformatics Bioinformatics applications." In: *Bioinformatics* (2002), pp. 1–56 (cit. on p. 17).

[WS09] K. Weinberger and L. Saul. "Distance Metric Learning for Large Margin Nearest Neighbor Classification." In: *Journal of Machine Learning Research* 10 (2009), pp. 207–244 (cit. on pp. 42, 43).

[Xi+06] Xiaopeng Xi et al. "Fast time series classification using numerosity reduction." In: *Proceedings of the 23rd international conference on Machine learning (ICML)*. 2006, pp. 1033–1040 (cit. on p. 16).

[YG08] J. Yin and M. Gaber. "Clustering distributed time series in sensor networks." In: *ICDM*. 2008 (cit. on p. 31).

[YL99] Yiming Yang and Xin Liu. "A re-examination of text categorization methods." In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval SIGIR 99*. 1999, pp. 42–49 (cit. on p. 17).

[G. 06] S. Thiria G. Dreyfus, J.-M. Martinez, M. Samuelides M. B. Gordon, F. Badran. *Apprentissage Apprentissage statistique*. Eyrolles. 2006, p. 471 (cit. on pp. 8, 11).

[Wie42] Wiener N. *Extrapolation, Interpolation & Smoothing of Stationary Time Series - With Engineering Applications*. Tech. rep. Report of the Services 19, Research Project DIC-6037 MIT, 1942 (cit. on p. 36).

Résumé — Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit. Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis.

Mots clés : Série temporelle, Apprentissage de métrique, k-NN, SVM, classification, régression.

Abstract — Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed non risus. Suspendisse lectus tortor, dignissim sit amet, adipiscing nec, ultricies sed, dolor. Cras elementum ultrices diam. Maecenas ligula massa, varius a, semper congue, euismod non, mi. Proin porttitor, orci nec nonummy molestie, enim est eleifend mi, non fermentum diam nisl sit amet erat. Duis semper. Duis arcu massa, scelerisque vitae, consequat in, pretium a, enim. Pellentesque congue. Ut in risus volutpat libero pharetra tempor. Cras vestibulum bibendum augue. Praesent egestas leo in pede. Praesent blandit odio eu enim. Pellentesque sed dui ut augue blandit sodales. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Aliquam nibh. Mauris ac mauris sed pede pellentesque fermentum. Maecenas adipiscing ante non diam sodales hendrerit. Ut velit mauris, egestas sed, gravida nec, ornare ut, mi. Aenean ut orci vel massa suscipit pulvinar. Nulla sollicitudin. Fusce varius, ligula non tempus aliquam, nunc turpis ullamcorper nibh, in tempus sapien eros vitae ligula. Pellentesque rhoncus nunc et augue. Integer id felis.

Keywords: Time series, Metric Learning, k-NN, SVM, classification, regression.

Schneider Electric
Université Grenoble Alpes, LIG
Université Grenoble Alpes, GIPSA-Lab