HAL
open science

Multiple Metric Learning for large margin kNN Classification of time series

Cao-Tri Do, Ahlame Douzal-Chouakria, Sylvain Marié, Michèle Rombaut

To cite this version:

Cao-Tri Do, Ahlame Douzal-Chouakria, Sylvain Marié, Michèle Rombaut. Multiple Metric Learning for large margin kNN Classification of time series. EUSIPCO 2015 - 23th European Signal Processing Conference, Aug 2015, Nice, France. hal-01202045

HAL Id: hal-01202045

https://hal.science/hal-01202045v1

Submitted on 30 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MULTIPLE METRIC LEARNING FOR LARGE MARGIN kNN CLASSIFICATION OF TIME SERIES

Cao-Tri Do[*†‡], *Ahlame Douzal-Chouakria*[†], *Sylvain Marié*[*], *Michèle Rombaut*[‡]

[*] Schneider Electric Industries
Grenoble
France

[†] Université Grenoble Alpes
CNRS-LIG/AMA
France

[‡] Université Grenoble Alpes
GIPSA-Lab/AGPiG
France

ABSTRACT

Time series are complex data objects, they may present noise, varying delays or involve several temporal granularities. To classify time series, promising solutions refer to the combination of multiple basic metrics to compare time series according to several characteristics. This work proposes a new framework to learn a combination of multiple metrics for a robust kNN classifier. By introducing the concept of pairwise space, the combination function is learned in this new space through a "large margin" optimization process. We apply it to compare time series on both their values and behaviors. The efficiency of the learned metric is compared to the major alternative metrics on large public datasets.

Index Terms— Multiple metric learning, Time series, kNN, Classification

1. INTRODUCTION

Nowadays, time series are present in various fields, particularly in emerging applications such as sensor networks, smart buildings, social media networks or Internet of Things [1–4]. Due to their temporal nature, time series constitute complex data to analyze by standard machine learning approaches [5].

To classify such challenging data, one may require advanced metrics bringing closer time series of identical classes while separating those of different classes. For this purpose, some propositions refer to embedding time series into new descriptive spaces such as spectral or tensor representations [6,7] or by using new dissimilarity metrics and temporal kernels [8–11]. As opposed to static data, temporal data may be compared not only on their values but also on their dynamics [12,13]. The most frequently used value-based metrics are the Euclidean distance [14] and the Dynamic Time Warping DTW to cope with delays [9,14,15].

Recent approaches show the benefit of combining multiple temporal metrics to classify challenging time series [12,13]. Such metrics are generally built as a combination of several behavior- and values-based metrics through a combination function set *a priori*, regardless of the analysis task [13]. In the same spirit than the metric learning approach introduced in Weinberger & Saul [16], we propose a new approach to learn a combined metric for a robust kNN classifier. The main idea is first to embed pairs of time series in a space whose dimensions are basic temporal metrics, and then to learn the metric combination function through a "large margin" optimization process in this space. The rest of the paper is organized as follows. Section 2 recalls briefly the major metrics for time series. In Section 3, we present the proposed multiple metric learning approach. Finally, Section 4 presents the experiments conducted and discusses the results obtained.

2. TIME SERIES METRICS

Let $\mathbf{x}_i = (x_{i1}, ..., x_{iT})$ and $\mathbf{x}_j = (x_{j1}, ..., x_{jT})$ be two time series of time length T. Generally, time series are asynchronous (i.e. varying delays), they need to be aligned before any comparison or analysis process. An alignment $\boldsymbol{\pi}_{ij}$ of length $|\boldsymbol{\pi}_{ij}| = M_{ij}$ (with $T \leq M_{ij} \leq 2T-1$) between two time series $\mathbf{x}_i$ and $\mathbf{x}_j$ is defined as the sequence of M_{ij} index pairs (π_i, π_j) of aligned elements in $\mathbf{x}_i$ and $\mathbf{x}_j$:

$$\boldsymbol{\pi}_{ij} = ((\pi_i(1), \pi_j(1)), ..., (\pi_i(t), \pi_j(t)), ..., (\pi_i(M_{ij}), \pi_j(M_{ij})))$$

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

$$1 = \pi_i(1) \leq \pi_i(2) \leq ... \leq \pi_i(M_{ij}) = T$$
$$1 = \pi_j(1) \leq \pi_j(2) \leq ... \leq \pi_j(M_{ij}) = T$$

and $\forall t \in \{1, ..., M_{ij}\}$,

$$\pi_i(t+1) \leq \pi_i(t) + 1 \text{ and } \pi_j(t+1) \leq \pi_j(t) + 1$$
$$(\pi_i(t+1) - \pi_i(t)) + (\pi_j(t+1) - \pi_j(t)) \geq 1$$

Let A be the set of all possible alignments. To find an optimal alignment, the Dynamic Time Warping algorithm

and its variants have been proposed [5, 15]. Given a cost function C (e.g. a dissimilarity measure), it computes the optimal alignment $\boldsymbol{\pi}^*_{ij}$ such that:

$$\boldsymbol{\pi}^*_{ij} = \underset{\pi_{ij} \in A}{\arg\min} \left(\sum_{t=1}^{M_{ij}} C(\mathbf{x}_{i\pi_i(t)}, \mathbf{x}_{j\pi_j(t)}) \right) \quad (1)$$

In the following, we consider the Euclidean distance as the cost function C. We suppose that the best alignment $\boldsymbol{\pi}^*_{ij}$ is found for each pair and note for simplification purpose $(\mathbf{x}_i, \mathbf{x}_j) = ((x_{i1}, ..., x_{it}, ..., x_{iM_{ij}}), (x_{j1}, ..., x_{jt}, ..., x_{jM_{ij}}))$ the time series after being aligned according to $\boldsymbol{\pi}^*_{ij}$.

Time series metrics fall at least within two main categories. The first one concerns value-based metrics (d_V), where time series are compared according to their values regardless of their behaviors. Among these metrics are the Euclidean distance (D_E), the Minkowski distance and the Mahalanobis distance [5].

The second category of metrics aims to compare time series based on their behavior regardless of the range of their values. By similar behavior, it is generally meant that for all periods $[t, t']$, the two time series increase or decrease simultaneously with the same growth rate in absolute value. On the contrary, they are said of opposite behavior if for all $[t, t']$, if one time series increases, the other one decreases (and vice-versa) with the same growth rate. Finally, time series are considered of different behaviors if they are not similar, nor opposite. Many applications refer to the Pearson correlation [12, 17] for behavior comparison. A generalization of the Pearson correlation has been introduced by Douzal & al. in [13]:

$$Cort_r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum_{t,t'} (x_{it} - x_{it'})(x_{jt} - x_{jt'})}{\sqrt{\sum_{t,t'} (x_{it} - x_{it'})^2} \sqrt{\sum_{t,t'} (x_{jt} - x_{jt'})^2}} \quad (2)$$

where $|t - t'| \leq r$, $r \in [1, ..., M_{ij} - 1]$ being a parameter that can be learned or fixed *a priori*. For $r = M_{ij} - 1$, Eq. 2 leads to the Pearson correlation. As $Cort_r$ is a similarity measure, it is transformed into a dissimilarity measure: $d_{B_r}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(1 - Cort_r(\mathbf{x}_i, \mathbf{x}_j))$.

Some applications show the benefit of involving both behavior and value (d_V) components through a combination function. A sigmoid combination function is proposed in [8, 13]:

$$D_{Sig}(\mathbf{x}_i, \mathbf{x}_j) = \frac{2d_V(\mathbf{x}_i, \mathbf{x}_j)}{1 + \exp(\alpha Cort_r(\mathbf{x}_i, \mathbf{x}_j))} \quad (3)$$

More generally, value (d_V) and behavior (d_{B_r}) dissimilarity metrics may be combined through a linear or geometric function:

$$D_{Lin}(\mathbf{x}_i, \mathbf{x}_j) = \alpha d_{B_r}(\mathbf{x}_i, \mathbf{x}_j) + (1 - \alpha) d_V(\mathbf{x}_i, \mathbf{x}_j) \quad (4)$$

$$D_{Geom}(\mathbf{x}_i, \mathbf{x}_j) = (d_{B_r}(\mathbf{x}_i, \mathbf{x}_j))^{\alpha} (d_V(\mathbf{x}_i, \mathbf{x}_j))^{1-\alpha} \quad (5)$$

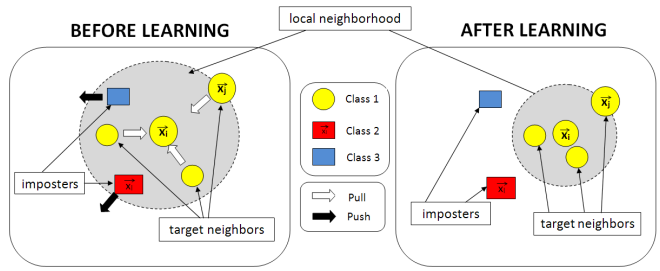

Fig. 1: Pushed and pulled samples in the $k = 3$ target neighborhood of $\mathbf{x}_i$ before (left) and after (right) learning. The pushed (vs. pulled) samples are indicated by a white (vs. black) arrows (Weinberger & Sault [16]).

α defines the value/behavior trade-off and can be learned through a grid search procedure. Note that these combination functions suffer from some limitations: they are fixed *a priori*, defined regardless of the analysis task, and are limited to two basic metrics. Our aim in this paper is to propose a new framework to learn a combined temporal metric D that combines several basic metrics for a robust kNN. In the next section, we first recall the metric learning framework proposed by Weinberger & Sault [16]. Then, we detail how this work is extended to multiple metric learning and applied to temporal data.

3. MULTIPLE METRIC LEARNING FOR A LARGE MARGIN *k*NN

3.1. Metric learning for a robust *k*NN classifier

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 & Sault proposed in [16] an approach to learn a dissimilarity metric D for a large margin kNN. It 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, they introduced the concept of *target* and *imposters* for each training sample $\mathbf{x}_i$. *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 $\mathbf{x}_i$. The *target* neighborhood is defined with respect to an initial metric; the learned metric D pulls the *targets* and pushes the *imposters* as shown in Figure 1.

In the following, we extend this framework to learn a combined metric for a large margin kNN.

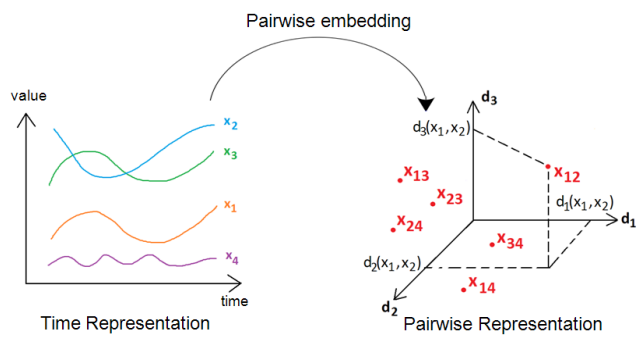

Fig. 2: Example of embedding of time series $\mathbf{x}_i$ from the temporal space (left) into the pairwise space (right). In this example, a pair of time series $(\mathbf{x}_1, \mathbf{x}_2)$ is projected into the pairwise space as a vector $\mathbf{x}_{12}$ described by $p = 3$ basic metrics: $\mathbf{x}_{12} = [d_1(\mathbf{x}_1, \mathbf{x}_2), d_2(\mathbf{x}_1, \mathbf{x}_2), d_3(\mathbf{x}_1, \mathbf{x}_2)]^T$.

3.2. Metric combination and pairwise space

Let $d_1, ..., d_h ..., d_p$ be p given dissimilarity metrics that allow to compare samples. The computation of a metric always takes into account a pair of samples. We introduce a new space representation referred as the pairwise space. In this new space, illustrated in Figure 2, a vector $\mathbf{x}_{ij}$ represents a pair of samples $(\mathbf{x}_i, \mathbf{x}_j)$ described by the p basics metrics d_h: $\mathbf{x}_{ij} = [d_1(\mathbf{x}_i, \mathbf{x}_j), ..., d_p(\mathbf{x}_i, \mathbf{x}_j)]^T$. If $\mathbf{x}_{ij} = \mathbf{0}$ then $\mathbf{x}_j$ is identical to $\mathbf{x}_i$ according to all metrics d_h.

A combination function D of the metrics d_h can be seen as a function in this space. We propose in the following to use a linear combination of d_h: $D_w(\mathbf{x}_i, \mathbf{x}_j) = \sum_h w_h.d_h(\mathbf{x}_i, \mathbf{x}_j)$. Its pairwise notation is:

$$D_w(\mathbf{x}_{ij}) = \mathbf{w}^T.\mathbf{x}_{ij} \tag{6}$$

3.3. Multiple metric learning in the pairwise space for a large margin kNN

The Weinberger & Sault framework can be used within the pairwise space to learn the combined metric D_w. The main steps of the proposed approach, detailed hereafter, can be summarized as follows:

1. Embed each pair $(\mathbf{x}_i, \mathbf{x}_j)$ into the pairwise space $\mathbb{R}^p$ as explained in Section 3.2.
2. Scale the data within the pairwise space.
3. Define for each $\mathbf{x}_i$ its *targets*.
4. Scale the neighborhood of each $\mathbf{x}_i$.
5. Learn the combined metric D_w.

Data scaling. 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.

Target set. For each $\mathbf{x}_i$, we define its *target* neighbors as the k nearest neighbors $\mathbf{x}_j$ $(j \rightsquigarrow i)$ of the same class according to an initial metric. In this paper, we choose a L2-norm of the pairwise space as an initial metric $(\sqrt{\sum_h d_h^2})$. Other metrics could be chosen. We emphasize that *target* neighbors are fixed *a priori* (at the first step) and do not change during the learning process.

Neighborhood scaling. In real datasets, local neighborhoods can have very different scales. To make the target neighborhood spreads comparable, we propose for each $\mathbf{x}_i$ to scale its neighborhood vectors $\mathbf{x}_{ij}$ such that the L2-norm of the farthest *target* is 1.

Learning the combined metric D_w. Let $\{\mathbf{x}_{ij}, y_{ij}\}_{i,j=1}^N$ be the training set with $y_{ij} = -1$ if $y_j = y_i$ and $+1$ otherwise. Learning D_w for a large margin kNN classifier can be formalized as the following optimization problem:

$$\begin{aligned}
&\min_{w,\xi} \underbrace{\sum_{i,j\rightsquigarrow i} D_w(\mathbf{x}_{ij})}_{pull} + \underbrace{C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl}}_{push} \\
&\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i, \\
&D_w(\mathbf{x}_{il}) - D_w(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\
&\xi_{ijl} \geq 0 \\
&w_h > 0 \forall h = 1...p
\end{aligned} \tag{7}$$

Note that the "pull" term $\sum_{j\rightsquigarrow i} D_w(\mathbf{x}_{ij}) = \sum_{j\rightsquigarrow i} \mathbf{w}^T.\mathbf{x}_{ij} = N.k.\mathbf{w}^T.\bar{\mathbf{x}}_{ij}$ is a L1-Mahalanobis norm weighted by the average target sample. Therefore, it behaves like a L1-norm in the optimization problem. The problem is very similar to a C-SVM classification problem. When C is infinite, we have a "strict" problem: the solver will try to find a direction in the pairwise space for which there is both no *imposters* invading the *target* spaces of each $\mathbf{x}_i$, and a maximum margin $\frac{1}{||\mathbf{w}||_2}$.

kNN classification of a new sample $\mathbf{x}_{test}$. Let $\mathbf{x}_{test,i}$ $(i = 1, ..., N)$ be the induced vectors in the pairwise space. These vectors are normalized according to the parameters retained at the *Data Scaling* step. $\mathbf{x}_{test}$ is classified with the kNN using the learned metric D_w.

4. EXPERIMENTATION

In this section, we compare kNN classifier performances for several temporal metrics on reference time series datasets [18–20] described in Table 1. To compare our results with the reference results in [14, 18], the experiments are conducted with the same protocols as in

Dataset	Nb. Class	Nb. Train	Nb. Test	TS length
SonyAIBO	2	20	601	70
MoteStrain	2	20	1252	84
GunPoint	2	50	150	150
PowerCons	2	73	292	144
ECG5Days	2	23	861	136
ECG200	2	100	100	96
SonyAIBOII	2	27	953	65
Coffee	2	28	28	286
BME	3	48	102	128
UMD	3	48	92	144

Table 1: Dataset description giving the number of classes (Nb. Class), the number of time series for the training (Nb. Train) and the testing (Nb. Test) sets, and the length of each time series (TS length).

Method	Parameter	Parameter range
d_{B_r}	r	$[1, 2, 3, , ..., T]$
D_{Lin}, D_{Geom}	α	$[0, 0.1, 0.2, ..., 1]$
D_{Sig}	α	$[0, 1, 2, ..., 6]$
D_2, D_8, D_{All}	C	$[10^{-4}, 10^{-3}, ..., 10^8]$

Table 2: Parameter ranges

Keogh &. Al: k is set to 1 ; train and test set are given a priori.

First, we solely consider basic metrics $d_V(\mathbf{x}_i, \mathbf{x}_j) = D_E(\mathbf{x}_i, \mathbf{x}_j)$ and $d_{B_r} = \frac{1}{2}(1 - Cort_r(\mathbf{x}_i, \mathbf{x}_j))$. Then, we consider linear (D_{Lin}, Eq. 4), geometric (D_{Geom}, Eq. 5) and sigmoid (D_{Sig}, Eq. 3) combination functions. For D_{Lin} and D_{Geom}, the dissimilarity metrics are Z-normalized on their log distributions as explained in Section 3.3. Finally, we learn the combined metric D_w according to the procedure described in 3.3. First, two basic metrics are considered in D_2 (d_V and d_{B_r}). Second, eight basic metrics are used for D_8: $d_1 = d_V$, $d_2, ..., d_8 = d_{B_r}$ based on $Cort_r$ for $r = 1, 0.05T, 0.1T, 0.2T, 0.25T, T, r^*$. Finally, in D_{All}, we consider d_V and d_{B_r} based on all orders of $Cort_r$. GNU Linear Programming Kit (GLPK Reference Manual [21]) is used to solve the optimization problem in Eq. 7 to find D_2, D_8, D_{All}.

Most of these metrics have parameters. We learn the optimal parameter values by minimizing a leave-one out cross-validation criterion. As the training dataset sizes are small, we propose a hierarchical error criterion:

1. Minimize the kNN error rate
2. Minimize $\frac{d_{intra}}{d_{inter}}$ if several parameter values obtain the minimum kNN error.

where d_{intra} and d_{inter} stands respectively to the mean of all intraclass and interclass distances according to the metric at hand. Table 2 gives the range of the grid search considered for the parameters.

Two experiments are conducted: first, the raw series are used (Table 3) ; then, time series are aligned according to the optimal alignment found by the DTW algorithm 1 with an Euclidean distance as the cost function

Dataset	Metrics							
	Basic		*A priori* combined			Learned combined		
	d_V	d_{B_r}	D_{Lin}	D_{Geom}	D_{Sig}	D_2	D_8	D_{All}
SonyAIBO	0.305	0.308	0.308	0.308	0.293	0.308	**0.173***	**0.173***
MoteStrain	**0.121***	0.264	0.217	0.197	0.231	**0.145**	0.193	0.185
GunPoint	**0.087**	**0.113**	**0.113**	**0.113**	**0.093**	**0.113**	**0.067***	**0.093**
PowerCons	**0.370**	0.445	0.445	0.431	**0.421**	**0.387**	**0.425**	0.441
ECG5Days	0.203	**0.153**	**0.153**	**0.153**	0.184	**0.153**	**0.147***	**0.170**
ECG200	**0.120**	**0.070***	**0.070***	**0.070***	**0.100**	**0.070***	**0.100**	**0.100**
SonyAIBOII	**0.141***	**0.142**	**0.142**	**0.142**	**0.144**	**0.142**	**0.155**	**0.155**
Coffee	0.250	**0***	**0***	**0***	**0.071**	**0***	**0***	**0***
BME	0.128	**0.059***	**0.059***	**0.059***	**0.059***	**0.059***	**0.059***	**0.059***
UMD	**0.185***	**0.207**	**0.207**	**0.207**	**0.207**	**0.207**	**0.207**	**0.196**

Table 3: Experimental results (error rate) of 1NN classifier for different metrics without DTW.

Dataset	Metrics							
	Basic		*A priori* combined			Learned combined		
	d_V	d_{B_r}	D_{Lin}	D_{Geom}	D_{Sig}	D_2	D_8	D_{All}
SonyAIBO	**0.275**	0.343	0.343	0.343	**0.265**	**0.275**	**0.236***	**0.236***
MoteStrain	**0.165***	**0.171**	**0.171**	**0.171**	**0.174**	**0.173**	**0.187**	**0.167**
GunPoint	0.093	**0.027***	**0.027***	**0.027***	**0.047**	**0.027***	**0.027***	**0.027***
PowerCons	**0.401**	**0.400**	**0.401**	**0.401**	**0.397***	**0.401**	**0.404**	**0.408**
ECG5Days	**0.232**	**0.236**	**0.235**	**0.235**	**0.241**	**0.236**	**0.229**	**0.224***
ECG200	**0.230**	**0.190**	**0.180***	**0.180***	**0.220**	**0.230**	**0.210**	**0.210**
SonyAIBOII	**0.169***	**0.194**	**0.169***	**0.169***	**0.178**	**0.169***	**0.169***	**0.169***
Coffee	**0.179**	**0.143***	**0.179**	**0.179**	**0.179**	**0.143***	**0.143***	**0.143***
BME	**0.128**	**0.118***	**0.118***	**0.118***	**0.118***	**0.118***	**0.118***	**0.118***
UMD	**0.120**	**0.109***	**0.109***	**0.109***	**0.109***	**0.109***	**0.109***	**0.109***

Table 4: Experimental results (error rate) of 1NN classifier for different metrics with DTW.

(Table 4). For all reported results, the best one is indexed with a star and the ones significantly similar from the best one (Z-test at 10% risk) are in bold [22].

From Table 3, we can see that value d_V or behavior d_{B_r} metrics alone performs better one from the other depending on the dataset. One basic metric does not always reach the best performance. There is a need to combine both of them into one metric. D_2 always reaches the best performance of d_V and d_{B_r} or is equivalent to the best one on all datasets. The new approach allows to extend combination functions to many metrics without having to cope with additional parameters in grid search (D_8, D_{All}). Adding metrics allows to outperform the two basics metrics (SonyAIBO). However, using a large number of metrics does not always improve the results (PowerCons, ECG200, SonyAIBOII). This might be caused by the fact that the many d_{B_r} are highly correlated: this probably hides the information of d_V when the target sets are computed with the initial metric (L2-norm in the pairwise space). Note that for MoteStrain, most combined metrics (*a priori* and learned) attain poorer performances than the best one of d_V and d_{B_r}. This may be due to the drastic difference between the training and test sets (Table 1).

From Table 4, we observe that aligning time series through DTW allows to reach better performances than without DTW on some databases (SonyAIBO, GunPoint, UMD), even with basic metrics. In this experiment, D_2 allows to reach the best performances of d_V and d_{B_r} or is equivalent to the best one on all datasets. This confirms

the observations made for Table 3. We note that for D_{All}, the best performances are attained for 7 datasets.

5. CONCLUSION

In this paper, we proposed a new method to learn a combination of multiple metrics for a robust kNN. It is based on a large margin optimization process in the pairwise space. We tested it on time series data to combine value- and behavior-based metrics, with good results.

For future work, we are looking for some improvements. **First**, the choice of the initial metric is crucial. It has been set here as the L2-norm of the pairwise space but a different metric could provide better *target* sets. Otherwise, using an iterative procedure (reusing D_w to generate new *target* sets and learn D_w again) could be another solution. Experiments have also shown that including many redundant metrics could affect the learning process. **Second**, we note that the L1-norm on the "pull" term leads to sparcity. Changing it into a L2-norm could allow for non-sparse solutions and also extend the approach to non-linear metric combination functions thanks to the Kernel trick. The Kernel function will have to be chosen carefully to ensure that D_w is monotonous in the pairwise space. **Third**, the learned metric could be used in other classification algorithms (decision tree, support vector machine, etc.) in its metric form or in a Kernel form. **Finally**, we could extend this framework to multivariate, regression or clustering problems.

REFERENCES

[1] J. Yin and M. Gaber, "Clustering distributed time series in sensor networks," in *ICDM*, 2008.

[2] H. Najmeddine, A. Jay, P. Marechal, and S. Marié, "Mesures de similarité pour l'aide à l'analyse des données énergétiques de bâtiments," in *RFIA*, 2012.

[3] L. Nguyen, P. Wu, W. Chan, W. Peng, and Y. Zhang, "Predicting collective sentiment dynamics from time-series social media," in *WISDOM*, 2012.

[4] M. Díaz, G. Juan, O. Lucas, and A. Ryuga, "Big data on the internet of things: An example for the E-health," in *IMIS*, 2012.

[5] T.C. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, 2011.

[6] J. Caiado, N. Crato, and D. Peña, "A periodogram-based metric for time series classification," *Computational Statistics and Data Analysis*, 2006.

[7] S. Papadimitriou, J. Sun, and C. Faloutsos, "Streaming pattern discovery in multiple time-series," in *VLDB*, 2005.

[8] A. Douzal-Chouakria, A. Diallo, and F. Giroud, "A random-periods model for the comparison of a metrics efficiency to classify cell-cycle expressed genes," *Pattern Recognition Letters*, 2010.

[9] M. Cuturi, J.P. Vert, O. Birkenes, and T. Matsui, "A Kernel for Time Series Based on Global Alignments," in *IEEE ICASSP*, 2006.

[10] A. Douzal-Chouakria and P. Nagabhushan, "Adaptive dissimilarity index for measuring time series proximity," *Advances in Data Analysis and Classification*, 2007.

[11] C. Frambourg, A. Douzal-Chouakria, and E. Gaussier, "Learning Multiple Temporal Matching for Time Series Classification," *Advances in Intelligent Data Analysis XII*, 2013.

[12] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," *Noise Reduction in Speech Processing*, 2009.

[13] A. Douzal-Chouakria and C. Amblard, "Classification trees for time series," *Pattern Recognition journal*, 2011.

[14] H. Ding, G. Trajcevski, and P. Scheuermann, "Querying and Mining of Time Series Data : Experimental Comparison of Representations and Distance Measures," in *VLDB*, 2008.

[15] H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE transactions on acoustics, speech, and signal processing*, 1978.

[16] K. Weinberger and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, 2009.

[17] Z. Abraham and P.N. Tan, "An Integrated Framework for Simultaneous Classification and Regression of Time-Series Data," in *ACM SIGKDD*, 2010.

[18] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C.A. Ratanamahatana, "The UCR Time Series Classification/Clustering Homepage (www.cs.ucr.edu/˜eamonn/time_series_data/)," 2011.

[19] K. Bache and M. Lichman, "UCI Machine Learning Repository (http://archive.ics.uci.edu/ml)," 2013.

[20] "LIG-AMA Machine Learning Datasets Repository (http://ama.liglab.fr/resourcestools/datasets/)," 2014.

[21] A. Makhorin, "GLPK, GNU Linear Programming Kit (https://www.gnu.org/software/glpk/)," 2006.

[22] T. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," 1997.