Proceedings 1st International Workshop on Advanced Analytics and Learning on Temporal Data
AALTD 2015

Temporal and Frequential Metric Learning for Time Series *k*NN Classification

Cao-Tri Do[123], Ahlame Douzal-Chouakria[2], Sylvain Marié[1], and Michèle Rombaut[3]

[1] Schneider Electric, France
[2] LIG, University of Grenoble Alpes, France
[3] GIPSA-Lab, University of Grenoble Alpes, France

Abstract. This work proposes a temporal and frequential metric learning framework for a time series nearest neighbor classification. For that, time series are embedded into a pairwise space where a combination function is learned based on a maximum margin optimization process. A wide range of experiments are conducted to evaluate the ability of the learned metric on time series kNN classification.

Keywords: Metric learning, Time series, kNN, Classification, Spectral metrics.

1 Introduction

Due to their temporal and frequential nature, time series constitute complex data to analyze by standard machine learning approaches [1]. In order to classify such challenging data, distance features must be used to bring closer time series of identical classes and separate those of different classes. Temporal data may be compared on their values. The most frequently used value-based metrics are the Euclidean distance and the Dynamic Time Warping DTW to cope with delays [2, 3]. They can also be compared on their dynamics and frequential characteristics [4, 5]. Promising approaches aims to learn the Mahalanobis distance or kernel function for a specific classifier [6, 7]. Other work investigate the representation paradigm by representating objects in a dissimilarity space where are investigated dissimilarity combinations and metric learning [8, 9]. The idea in this paper is to combine basic metrics into a discriminative one for a kNN classifier. In the metric learning context for a metric learning approach driven by nearest neighbors (Weinberger & Saul [6]), we extend the work of Do & al. in [10] to temporal and frequential characteristics. The main idea is to embed pairs of time series in a space whose dimensions are basic temporal and frequential metrics, where a combination function is learned based on a large margin optimization process.

The main contributions of the paper are a) propose a new temporal and frequential metric learning framework for a time series nearest neighbors classification, b) learn a combination metric involving amplitude, behavior and frequential characteristics and c) conduct large experimentations to study the ability of learned metric. 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

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. Time series metrics fall at least within three main categories. The first one concerns value-based metrics, 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 [3]. We recall the formula of d_E:

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

The second category relies on metrics in the spectral representations. In some applications, time series may be similar because they share the same frequency characteristics. For that, time series $\mathbf{x}_i$ are first transformed into their Fourier representation $\tilde{\mathbf{x}}_i = [\tilde{x}_{i1}, ..., \tilde{x}_{iF}]$, where $\tilde{x}_{if}$ is a complex number (i.e. Fourier components), with $F = \frac{2^T}{2} + 1$ [5]. Then, one may use the Euclidean distance (d_{FFT}) between the module of the complex numbers $\tilde{x}_{if}$, noted $|\tilde{x}_{if}|$:

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

Note that times series of similar frequential characteristics may have distinctive global behavior. Thus, to compare time series based on their behavior, a third category of metrics is used. Many applications refer to the Pearson correlation or its generalization, the temporal correlation coefficient [4] defined as:

$$Cort_r(\mathbf{x}_i, \mathbf{x}_j) = \frac{\sum\limits_{t,t'}(x_{it} - x_{it'})(x_{jt} - x_{jt'})}{\sqrt{\sum\limits_{t,t'}(x_{it} - x_{it'})^2}\sqrt{\sum\limits_{t,t'}(x_{jt} - x_{jt'})^2}} \tag{3}$$

where $|t - t'| \leq r$, $r \in [1, ..., T-1]$ being a parameter that can be learned or fixed *a priori*. The optimal value of r is noisy dependant. For $r = T - 1$, Eq. 3 leads to the Pearson correlation. As $Cort_r$ is a similarity measure, it is transformed into a dissimilarity measure: $d_{Cort_r}(\mathbf{x}_i, \mathbf{x}_j) = \frac{1}{2}(1 - Cort_r(\mathbf{x}_i, \mathbf{x}_j))$.

3 Temporal and frequential metric learning for a large margin *k*NN

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 [6] 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* 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 ($y_j = y_i$). The *target* neighborhood is defined with respect to an initial metric. The aim is to learn a metric D that pulls the *targets* and pushes the ones of different class.

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. Therefore, we used the pairwise representation introduced in Do & al. [10]. In this space, 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 $D_w(\mathbf{x}_{ij}) = \mathbf{w}^T.\mathbf{x}_{ij}$. To ensure that D_w is a valid metric, we set $w_h \geq 0$ for all $h = 1...p$. The main steps of the proposed approach to learn the metric, detailed hereafter, can be summarized as follows:

1. Embed each pair $(\mathbf{x}_i, \mathbf{x}_j)$ into the pairwise space $\mathbb{R}^p$.
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 the 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:

$$\begin{aligned} &\min_{w,\xi} \underbrace{\sum_{i,j\rightsquigarrow i} D_w(\mathbf{x}_{ij})}_{pull} + \underbrace{C \sum_{i,j\rightsquigarrow i,l} \frac{1+y_{il}}{2}.\xi_{ijl}}_{push} \\ &\text{s.t. } \forall j \rightsquigarrow i, y_l \neq y_i, \\ &D_w(\mathbf{x}_{il}) - D_w(\mathbf{x}_{ij}) \geq 1 - \xi_{ijl} \\ &\xi_{ijl} \geq 0 \\ &w_h > 0 \ \forall h = 1...p \end{aligned} \tag{4}$$

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 only targets are in the close neighborhood of each $\mathbf{x}_i$, and a maximum margin $\frac{1}{||\mathbf{w}||_2}$.

Let $\mathbf{x}_{test}$ be a new sample to classify and $\mathbf{x}_{test,i}$ $(i = 1, ..., N)$ the corresponding vectors into the pairwise embedding space. After $\mathbf{x}_{test,i}$ normalization according to the *Data Scaling* step, $\mathbf{x}_{test}$ is classified based on a standard kNN and D_w.

4 Experiments

In this section, we compare kNN classifier performances for several metrics on reference time series datasets [11–14] described in Table 1. To compare with the reference results in [3, 11], the experiments are conducted with the same protocols as in Do &. al. [10]: k is set to 1; train and test set are given a priori. Due to the current format to store the data, small datasets with short time series were retained and the experiments are conducted on one runtime.

In this experimentation, we consider basic metrics d_E, d_{FFT} and d_{Cort_r} then, we learn a combined metric D_w according to the procedure described in Section 3. First, two basic temporal metrics are considered in D_2 (d_E and d_{Cort_r}) as in Do & al. [10]. Second, we consider a combination between temporal and frequential metrics in D_3 (d_E, d_{Cort_r} and d_{FFT}). Cplex library [15] has been used to solve the optimization problem in Eq. 4. We learn the optimal parameter values of these metrics 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. In the following, we consider only the raw series and don't align them using a DTW algorithm for example. For

all reported results (Table 3), the best one is indexed with a star and the ones significantly similar from the best one (Z-test at 1% risk) are in bold [16].

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
SonyAIBOII	2	27	953	65
Coffee	2	28	28	286
BME	3	48	102	128
UMD	3	46	92	150
ECG200	2	100	100	96
Beef	5	30	30	470
DiatomSizeReduction	4	16	306	345
FaceFour	4	24	88	350
Lighting-2	2	60	61	637
Lighting-7	7	70	73	319
OliveOil	4	30	30	570

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_{Cort_r}	r	$[1, 2, 3, , ..., T]$
D_2, D_3	C	$[10^{-3}, 0.5, 1, 5, 10, 20, 30, ..., 150]$

Table 2. Parameter ranges

From Table 3, we can see that temporal metrics d_E and d_{Cort_r} alone performs better one from the other depending on the dataset. Using a frequential metric alone such as d_{FFT} brings significant improvements for some datasets (SonyAIBO, GunPoint, PowerCons, ECG5Days). It can be observed that one basic metric is sufficient on some databases (MoteStrain, GunPoint, PowerCons, ECG5Days). In other cases, learning a combination of these basic metrics reach the same performances on most datasets or achieve better results (UMD). The new approach allows to extend combination functions to many metrics without having to cope with additional parameters in grid search and without to test every basic metrics alone to retained the best one. It also extends the work done in [6] for single distance to multiple distances. Adding metrics such as d_{FFT} improves the performances on some datasets (SonyAIBO, GunPoint, UMD, FaceFour, Lighting-2, Lighting-7) than considering only temporal metrics

Dataset	Metrics				
	Basic			Learned combined	
	d_E	d_{Cort_r}	d_{FFT}	D_2	D_3
SonyAIBO	0.305	0.308	**0.258***	0.308	**0.259**
MoteStrain	**0.121***	0.264	0.278	0.210	0.277
GunPoint	0.087	0.113	**0.027***	0.113	0.073
PowerCons	**0.370**	0.445	**0.315***	0.384	0.410
ECG5Days	0.203	0.153	**0.006***	0.153	0.156
SonyAIBOII	**0.141**	**0.142**	**0.128***	**0.142**	**0.142**
Coffee	0.250	**0***	0.357	**0***	**0***
BME	0.128	**0.059***	0.412	**0.059***	**0.078**
UMD	**0.185***	0.207	0.315	0.207	**0.185***
ECG200	0.120	**0.070***	0.166	**0.070***	**0.070***
Beef	**0.467**	**0.300***	**0.500**	**0.300***	**0.367**
DiatomSizeReduction	**0.065***	**0.075**	**0.069**	**0.075**	**0.075**
FaceFour	**0.216**	**0.216**	**0.239**	**0.216**	**0.205***
Lighting-2	**0.246**	**0.246**	**0.148***	**0.246**	**0.213**
Lighting-7	**0.425**	**0.411**	**0.315**	**0.411**	**0.288***
OliveOil	**0.133***	**0.133***	**0.200**	**0.133***	**0.133***

Table 3. Error rate of 1NN classifier for different metrics. D_2 is computed using d_E and d_{Cort_r}; D_3 uses the 3 basic metrics. The metric with the best performance for each dataset is indicated by a star (*) and the ones with equivalent performances are in bold.

(d_E, d_{Cort_r}). However, it does not always improve the results (GunPoint, PowerCons, ECG5Days). This might be caused by the fact that our framework is sensitive to the choice of the initial metric (L2-norm) or maybe, some steps in the algorithm should be improved to make the combination better.

5 Conclusion

For nearest neighbor time series classification, we propose to learn a metric as a combination of temporal and frequential metrics based on a large margin optimization process. The learned metric shows good performances on the conducted experimentations. 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. **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. **Finally**, we could extend this framework to multivariate, regression or clustering problems.

References

1. T.C. Fu, "A review on time series data mining," *Engineering Applications of Artificial Intelligence*, 2011.
2. H. Sakoe and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition," *IEEE transactions on acoustics, speech, and signal processing*, 1978.
3. H. Ding, G. Trajcevski, and P. Scheuermann, "Querying and Mining of Time Series Data : Experimental Comparison of Representations and Distance Measures," in *VLDB*, 2008.
4. A. Douzal-Chouakria and C. Amblard, "Classification trees for time series," *Pattern Recognition journal*, 2011.
5. S. Lhermitte, J. Verbesselt, W.W. Verstraeten, and P. Coppin, "A comparison of time series similarity measures for classification and change detection of ecosystem dynamics," 2011.
6. K. Weinberger and L. Saul, "Distance Metric Learning for Large Margin Nearest Neighbor Classification," *Journal of Machine Learning Research*, 2009.
7. M. Gönen and E. Alpaydin, "Multiple kernel learning algorithms," *The Journal of Machine Learning Research*, vol. 12, pp. 2211–2268, 2011.
8. R. Duin, M. Bicego, M. Orozco-alzate, S. Kim, and M. Loog, "Metric Learning in Dissimilarity Space for Improved Nearest Neighbor Performance," pp. 183–192.
9. A. Ibba, R. Duin, and W. Lee, "A study on combining sets of differently measured dissimilarities," in *Proceedings - International Conference on Pattern Recognition*, 2010, pp. 3360–3363.
10. C. Do, A. Douzal-Chouakria, S. Marié, and M. Rombaut, "Multiple Metric Learning for large margin kNN Classification of time series," *EUSIPCO*, 2015.
11. 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.
12. K. Bache and M. Lichman, "UCI Machine Learning Repository (http://archive.ics.uci.edu/ml)," 2013.
13. "LIG-AMA Machine Learning Datasets Repository (http://ama.liglab.fr/resourcestools/datasets/)," 2014.
14. C. Frambourg, A. Douzal-Chouakria, and E. Gaussier, "Learning Multiple Temporal Matching for Time Series Classification," *Advances in Intelligent Data Analysis XII*, 2013.
15. "IBM Cplex (http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/)," .
16. T. Dietterich, "Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms," 1997.