On-line Dynamic Time Warping for Streaming Time Series

Izaskun Oregi[1], Aritz Pérez[2], Javier Del Ser[1,2,3], and José A. Lozano[2,4]

[1] TECNALIA, 48160 Derio, Spain
{izaskun.oregui,javier.delser}@tecnalia.com
[2] Basque Center for Applied Mathematics (BCAM), 48009 Bilbao, Spain
{aperez,jdelser}@bcamath.org
[3] Department of Communications Engineering
University of the Basque Country UPV/EHU, 48013 Bilbao, Spain
javier.delser@ehu.eus
[4] Department of Computer Science and Artificial Intelligence
University of the Basque Country UPV/EHU, 20018 Donostia-San Sebastián, Spain
ja.lozano@ehu.eus

Abstract. Dynamic Time Warping is a well-known measure of dissimilarity between time series. Due to its flexibility to deal with non-linear distortions along the time axis, this measure has been widely utilized in machine learning models for this particular kind of data. Nowadays, the proliferation of streaming data sources has ignited the interest and attention of the scientific community around on-line learning models. In this work, we naturally adapt Dynamic Time Warping to the on-line learning setting. Specifically, we propose a novel on-line measure of dissimilarity for streaming time series which combines a warp constraint and a weighted memory mechanism to simplify the time series alignment and adapt to non-stationary data intervals along time. Computer simulations are analyzed and discussed so as to shed light on the performance and complexity of the proposed measure.

Keywords: Time series, on-line learning, Dynamic Time Warping

1 Introduction

In many fields such as manufacturing industry, energy, finance or health, time series are one of the most common forms under which data are captured and processed towards extracting valuable information. For this purpose, time series classification has played a central role in time series analysis: the goal is to build a predictive model based on labeled time series so as to use it to predict the label of previously unseen, unlabeled time series. In the presence of labeled data, k-Nearest Neighbor (k-NN) classification models have been extensively utilized with time series data due to their conceptual simplicity, efficiency and ease of implementation. In essence, k-NN algorithms consist of assigning a label to an unseen example according to the class distribution over its k most

similar (*nearest*) data instances within the training set. It is obvious that the accuracy of nearest-neighbor techniques is closely related to the measure of similarity between examples. In this regard, research in pattern recognition for time series has originated a diverse collection of measures including the Euclidean Distance (ED), Elastic Similarity Measures (ESM) and Longest Common Subsequence (LCSS), each featuring properties and limitations that should match the requirements of the application at hand.

To the best of our knowledge, no attention has been paid in the literature to distance-based on-line classification models for time series data streams that build upon a proper design of elastic measures of similarity. In response to this lack of research, this work elaborates on an on-line DTW (ODTW) dissimilarity measure. The fundamental ingredient of the ODTW is given by a spotted DTW property that is exploited to avoid unnecessary dissimilarity recalculations. Moreover, computational resources (time and memory) are also controlled by virtue of a Sakoe-Chiba bounding approach. Finally, by under-weighting the influence of past events using a weighted memory mechanism, we make it possible to adapt the ODTW to non-stationarities in the data stream, in clear connection with the well-known stability-plasticity dilemma in on-line learning models. In order to assess the practical performance of the proposed ODTW dissimilarity measure under changing classification concepts, extensive experiments using 1-NN classifiers will be discussed over different public datasets. The ODTW accuracy rate as new data samples arrive will be compared to that of the DTW measure, showing that ODTW can be at least as accurate as DTW. The efficiency of the ODTW in terms of complexity will be also analyzed.

The remainder of the paper is organized as follows: Section 2 provides background information on time series similarity measures. Section 3 formulates the definition of the conventional DTW measure and introduces the Sakoe-Chiba band technique. Section 4 gives a detailed description of the proposed ODTW dissimilarity measure. Section 5 delves into the obtained experimental results and, finally, Section 6 summarizes the contributions and outlines future research lines, leveraging our findings in this work.

2 Background

Despite its neat advantages – low complexity and simplicity – ED is overly inflexible to deal with time series distortions. In classification problems where the learning process usually focuses on the shape of sequence, this limitation might pose a severe problem. To overcome this issue, Dynamic Time Warping (DTW), an elastic measure of similarity, has proved to be extremely effective to align sequences that are similar in shape but undergo non-linear variations in the time dimension. Along with DTW, the ESM family is completed by the so-called Edit distance [18], the Edit Distance for Real sequences (EDR, [3]) and the Edit distance with Real Penalty (EPR, [2]), among other DTW-based distances [12]. In general, the most important characteristic of all ESMs is their ability to shrink or stretch the time axis in order to find the alignment between the time

series under comparison yielding the smallest distance. The ground difference among them, conversely, lies in the selected point-wise distance. Similarly, LCSS is a variation of ESM techniques that allows instances to be unmatched, i.e., a global sequence alignment is not required. Several studies have shown that the use of ESM with 1-NN classifiers outperforms results wich are very accurate and hardly beaten in several classification problems [5, 15]. Standing on this empirical evidence, DTW-based models have consolidated as the reference for shape-based classification tasks over time series data [9, 17, 16, 23].

An important point to keep in mind when dealing with the DTW similarity measure is its quadratic computational complexity, which makes its computation prohibitive when tackling long time series. In order to avoid this drawback, techniques such as Itakura's Parallelogram [8] and Sakoe-Chiba band [20] are widely utilized to reduce the DTW complexity. These simple methods speed up the DTW computation just by limiting the flexibility of the measure when accommodating time-axis distortions. Similarly to these constraint-based methods, Salvador and Chan [22] estimate the DTW measure by means of a multi-level approach that recursively refines its resolution. Likewise, Keogh and Pazzani [11] propose a modified DTW approach which uses Piecewise Aggregate Approximation (PAA) in order to reduce the length of the time series under comparison and speed up the final computation. Indexing time series to accelerate the performance of different learning methods where DTW is involved is another solution to alleviate its computation [10, 21, 14].

The complexity issue of the DTW measure noted above is particularly challenging when time series are generated continuously along time, producing endless data streams potentially produced by non-stationary distributions. In many scenarios, data produced by systems and/or processes evolve over time, not necessarily in a stationary manner, making conventional classification methods unsuitable to handle data produced by time-varying generation processes. These stringent conditions under which stream data must be processed have motivated a recent upsurge of *on-line* classification models [13, 7, 24, 6]. In order to identify changes in time series data generator models, Cavalcante et al. [1] have recently proposed a concept drift detector method coined as FEDD. Based on the feature vector similarity given by Pearson correlation distance (or cosine distance), this method monitors the evolution of sequence features in order to test whether a concept change has occurred. In [19] an incremental clustering system for time series data streams is presented: On-line Divisive-Agglomerative Clustering is a tree-like grouping technique that evolves with data based on a criterion to merge and split clusters using a correlation-based dissimilarity measure.

3 Dynamic Time Warping

The DTW measure between two time series, $X^m = (x_1, \ldots, x_i, \ldots, x_m)$ and $Y^n = (y_1, \ldots, y_j, \ldots, y_n)$, is given by the minimum cumulative distance resulting from the best point-wise alignment between both time series.

We represent an alignment of two time series X^m and Y^n by a **path** $p = \{(i_1, j_1), ..., (i_Q, j_Q)\}$ that goes from $(1, 1)$ to (m, n) in a $[1, m]\times[1, n]$ lattice. Each pair $(i, j) \in p$ represents the alignment of the points x_i and y_j. We say that the path p is **allowed** if it satisfies that $(i_q, j_q) - (i_{q-1}, j_{q-1}) \in \{(1, 0), (1, 1), (0, 1)\}$ for $q = 2, ..., Q$. That is, allowed paths are formed by $\uparrow, \rightarrow$ and $\nearrow$ steps. Figure 1.a shows three possible paths (alignments) between time series X^m and Y^n. From here on, we will consider only allowed paths and, therefore, we will omit the term allowed, for the sake of brevity.

The **weight** of a path p is given by

$$w(p) = \sum_{(i,j)\in p} d_{i,j} \tag{1}$$

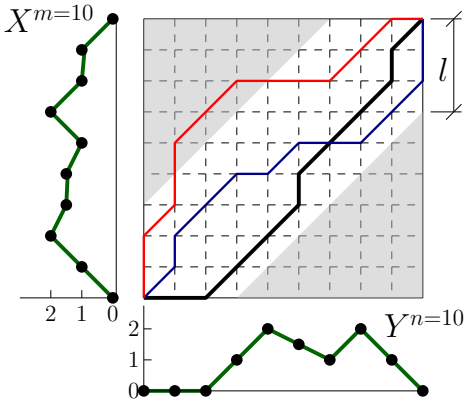
where $d_{i,j} = |x_i - y_j|$ is the point-wise distance. Next, we present the definition of the DTW measure:

Definition 1. *The **DTW** measure between time series X^m and Y^n is given by*

$$D(X^m, Y^n) = \min_{p\in\mathcal{P}} w(p)$$

where $\mathcal{P}$ is the set of all allowed paths in the $[1, m] \times [1, n]$ lattice.

When it is clear from the context, we will denote $D(X^m, Y^n)$ simply by $D_{m,n}$. The DTW value corresponds to the weight of the **optimal path**, i.e., the minimum weighted path.



(a) Time series alignment.

11	7	11	4	5	3.5	2	3	1	0
11	11	11	3	4	2	1	2	0	1
10	10	10	3	3	1.5	1	1	0	1
9	9	9	3	2	1	1	0	1	3
7	7	7	2	2	0.5	0	1	1	2
6	6	6	2	1	0	0.5	1	1.5	3
4.5	4.5	4.5	1.5	0.5	0	0.5	1	1.5	3
3	3	3	1	0	0.5	1.5	1.5	2.5	4.5
1	1	1	0	1	1.5	1.5	2.5	2.5	3.5
0	0	0	1	3	4.5	5.5	7.5	8.5	8.5

(b) Cumulative distance matrix.

Fig. 1. Example of the computation of $D(X^m, Y^n)$ for two time series. It can be observed that the optimal path $\mathbf{w}^*$ (—) is that yielding the minimum cumulative distance among those paths connecting points $(1, 1)$ to $(10, 10)$.

Since the number of allowed paths grows exponentially with the length of the time series X^m and Y^n, an exhaustive enumeration of all of them with the aim

of finding the optimal path is computationally unfeasible, even for small values of n and m. Fortunately, by using dynamic programming, we can compute the DTW measure between two time series using the following recursion:

$$D_{m,n} = d_{m,n} + \min\left\{D_{m-1,n}, D_{m-1,n-1}, D_{m,n-1}\right\}, \tag{2}$$

where $D_{0,0} = 0$ and $D_{i,0} = D_{0,j} = \infty$ for $i = 1, \ldots, m$ and $j = 1, \ldots, n$, respectively (initial conditions). Using this recurrence, the complexity for computing the DTW measure between time series X^m and Y^n is $\mathcal{O}(m \cdot n)$. This is still unaffordable in many practical situations and, therefore, several techniques have been proposed to approximate the DTW measure by decreasing its computational complexity. Many of these techniques are based on reducing the number of paths by imposing additional constraints. Among these, we would like to highlight the Sakoe-Chiba bound and Itakura's parallelogram approaches due to their effective constraints. These constraints allow i) discarding the subset of paths of higher lengths and (on average) with higher weights, and ii) computing the (approximated) DTW measure in linear time with respect to the length of the time series considered.

In particular, the paths considered by the Sakoe-Chiba bound approach are composed by pairs (i_q, j_q) that satisfy the constraints $|i_q - j_q| \leq l$ for $1 \leq q \leq Q$, where l refers to the so-called **band width**. We call these additional constraints to the paths as the **Sakoe-Chiba constraints**. As a consequence, the DTW can be computed in $\mathcal{O}(l \cdot \max\{m, n\})$. The areas shadowed in Figure 1.a illustrate the forbidden (i_q, j_q) pairs for $l = 3$. In this case, the red path is not allowed while the blue and black paths remain allowed.

4 On-line Dynamic Time Warping

In this section, we propose the on-line DTW (**ODTW**). The ODTW measure combines i) an incremental computation of the DTW measure, ii) the Sakoe-Chiba bound for limiting the computational and space complexities and iii) a weighted memory mechanism. By the combination of these ideas, ODTW can be computed efficiently (i and ii) and it can control the contribution of the past values to the measure. Next, we introduce the three ideas in order and, finally, we combine them into the novel ODTW.

4.1 Controlling the Computational Complexity

When computing $D_{m,n}$ using Equation (4), we also compute $D_{i,j}$ for $i = 1, ..., m$ and $j = 1, ..., n$. These values correspond to the DTW measures for time series X^i and Y^j for $1 \leq i \leq m$ and $1 \leq j \leq n$.

We call $M^{m,n} = \{D_{i,j} : i = 1, ..., m \text{ and } j = 1, ..., n\}$ the **measure matrix**. Figure 1.b shows the measure matrix obtained when computing $D_{10,10}$. The set of values of the measure matrix shaded in gray corresponds to the set of DTW measures in the optimal path p^* associated to $D_{10,10}$, i.e., $\{D_{i,j} : (i, j) \in p^*\}$.

We call $F^{m,n} = \{D_{i,n}, D_{m,j} : 1 \leq i \leq m, 1 \leq j \leq n\}$ the **frontier** of the measure matrix $M^{m,n}$. In Figure 1.b the frontier $F^{7,8}$ corresponds to the set of matrix values shaded in green (plus the value at position $(7, 8)$, shaded in gray).

Let us assume that we know the frontier $F^{r,s}$ for a given $1 \leq r < m$ and $1 \leq s < n$ and that we want to calculate $D_{m,n}$. Interestingly, in order to compute $D_{m,n}$, we can apply Equation (4) recursively until a value from the frontier $F^{r,s}$ needs to be computed. At this point, the recursion can be stopped, which can avoid unnecessary calculations (see Figure 2.a). This simple idea is the basis for an incremental computation of the DTW in an on-line scenario.

Now, imagine a (general) on-line scenario where the time series arrive in chunks, sequentially. For instance, at time t_0, we can have the time series X^r and Y^s and, then, at time t_1, we can receive $X^{r+1,m} = (x_{r+1}, ..., x_m)$ and $Y^{s+1,n} = (y_{s+1}, ..., y_m)$. We want to compute $D_{m,n}$ incrementally. At time t_0, we can compute $D_{r,s}$, store the frontier $F^{r,s}$, and the time series X^r and Y^s. Then, at time t_1, we can compute $D_{m,n}$ using the stored frontier according to the previously described mechanism (see Figure 2.a). This incremental process can be repeated when a new chunk arrives.

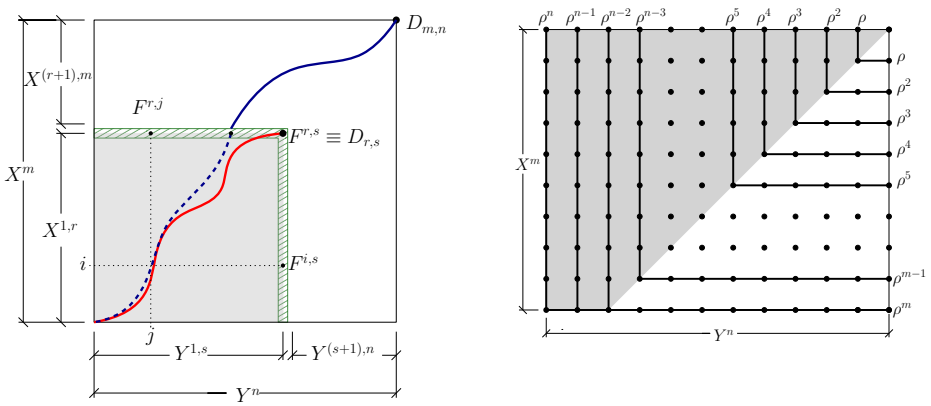
Given the frontier $F^{r,s}$, the computational complexity for obtaining $D_{m,n}$ and $F^{m,n}$ using the proposed incremental procedure is $\mathcal{O}(m \cdot n - r \cdot s)$. We would like to highlight that, when a single point arrives for both time series ($m = r + 1$ and $n = s + 1$), the computational complexity is linear with respect to the length of the time series, $\mathcal{O}(\max\{m, n\})$. In addition, the described procedure requires X^m and Y^n to be stored, which leads to a space complexity of $\mathcal{O}(\max\{m, n\})$.

Unfortunately, the computational and space complexities of the proposed incremental computation of the DTW measure are impractical for most of the challenging on-line scenarios. To overcome this issue, Sakoe-Chiba constraints (see end of Section 3) can be imposed to the incremental computation, drastically reducing both the memory store and the computational complexity.

By using a band width l, after computing $D_{r,s}$, we store only $\mathcal{O}(l)$ values of the frontier $F^{r,s}$, because some values could correspond to a pair of points that do not fulfill the Sakoe-Chiba constraints. In addition, we only need to store the last l points of the time series X^r and Y^s, which leads to a space complexity of $\mathcal{O}(l)$. In addition, the computational complexity for calculating incrementally an approximation to $D_{m,n}$ according to Sakoe-Chiba constraints is linear in l, i.e., $\mathcal{O}((m-r) \cdot (n-s) + l \cdot (m+n-r-s))$. Therefore, by choosing an appropriate band width l, the proposed constrained and incremental DTW can effectively control the trade-off between i) the required computational and memory resources, and ii) the flexibility of DTW with respect to the distortions of the time series along the time axis.

4.2 Forgetting the Past

One of the most extended assumptions in off-line learning is that data samples are drawn from a stationary distribution. However, in on-line learning scenarios this assumption may not hold as stationarity of the data streaming can evolve over time. In consequence, we conceive a streaming time series as being divided



(a) Incremental DTW computation. (b) Weights over the DTW lattice.

Fig. 2. (a) Example of the incremental computation of the DTW measure, and (b) exponential weights applied over a $[1, m] \times [1, n]$ lattice.

into a sequence of stationary intervals (concepts) of varying length and with a similar periodic shape. We say that a **concept drift** has occurred when the time series changes from one stationary interval to another.

In order to adapt the DTW measure to the concept drift phenomenon, we propose a simple yet efficient weighted memory mechanism that modulates the contribution of the point-wise distance $d_{i,j}$ in the weight of a path (see Equation (1)). This mechanism leverages the assumption that recent points are more likely to have been produced in the last stationary interval. To this end, we propose the use of a memory parameter $\rho \in (0, 1]$ and we define the weight with memory of a path p as follows:

$$w_\rho(p) = \sum_{(i,j)\in p} \rho^{\max\{m-i,n-j\}} d_{i,j} \tag{3}$$

Note that $\max\{m-i, n-j\}$ corresponds to the Chebyshev distance between the points (i, j) and (m, n). In this manner, points belonging to the last stationary interval of a time series will contribute more significantly to the weight of a given path and to a measure that minimizes this weight with memory.

Figure 2.b illustrates an example of the weighting function given in Equation 3 for $m = n - 2$. All the pairs (i, j) in the lattice $[1, m] \times [1, n]$ at the same Chebyshev distance from (m, n) are connected with a black line.

We would like to highlight that, contribution of a pair of points (x_i, y_j) to the memory weight of any path is equal. In other words, the contribution of a pair of points to the memory weight does not depend on the path, which avoids favoring the longest paths. Intuitively, an appropriate memory parameter ρ should be selected according to the length of the stationary intervals or pattern period (see Section 5)

4.3 The ODTW Measure

Next, we propose the definition of the on-line DTW measure (ODTW).

Definition 2 (On-line Dynamic Time Warping). *The **ODTW** measure between time series X^m and Y^n given the memory parameter ρ and the band width l is given by*

$$D_{l,\rho}(X^m, Y^n) = \min_{p \in P_l} w_\rho(p)$$

where P_l is the set of all the paths in the $[1, m] \times [1, n]$ lattice satisfying the Sakoe-Chiba constraints (see the end of Section 3).

From here on, when it is clear from the context, we will denote $D_{l,\rho}(X^m, Y^n)$ simply by $D_{m,n}$.

Note that the proposed ODTW can be understood as a generalization of the DTW measure. For instance, if $\rho = 1$ and $l = \infty$ – without Sakoe-Chiba constraints – then ODTW corresponds to DTW. Additionally, when $n = m$, $\rho = 1$ and $l = 0$ ODTW corresponds to the Euclidean distance.

Again, by using dynamic programming, it is possible to compute the ODTW measure given ρ and l between X^m and Y^n using the following recursion

$$D_{m,n} = d_{m,n} + \min \left\{ \rho^{\mathbb{I}(m>n)} \cdot D_{m-1,n}, \rho \cdot D_{m-1,n-1}, \rho^{\mathbb{I}(m<n)} D_{m,n-1} \right\}, \quad (4)$$

where $D_{i,j} = \infty$ for any pair (i, j) not satisfying the Sakoe-Chiba additional constraints, and $\mathbb{I}(\cdot)$ is an auxiliary function taking value 1 if its argument is true, and 0 otherwise.

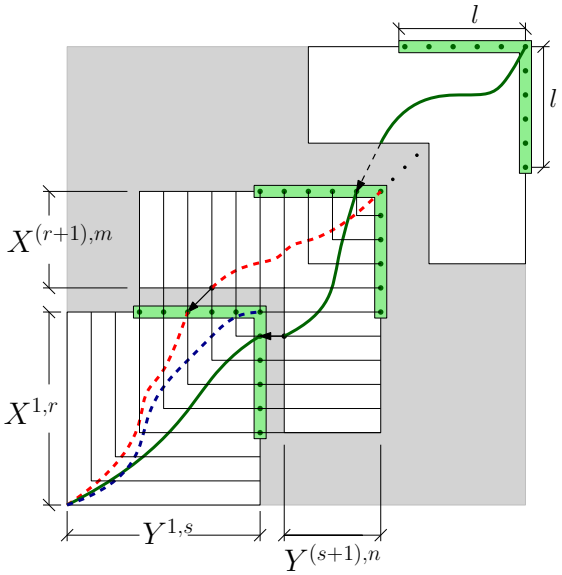


Fig. 3. Computation of the proposed ODTW measure in its general form.

At this point, we would like to mention that Definition 2 and its recursive computation shown in Equation (4) have been given in their simplest form, for the sake of brevity and readability. The simplest form is appropriate for on-line scenarios where the time series arrive in chunks consisting of a single point. However, both the definition and the recursive computation of ODTW can be easily generalized in order to deal with chunks of arbitrary sizes. Figure 3 illustrates the on-line DTW recursive computation in the general form, that is, when the time series arrive in chunks of arbitrary sizes. The green areas represent the stored l-frontiers.

Note: The source code in Python of the proposed ODTW measure (in its most general form) has been made available on-line at `http://bitbucket.org/izaskun_oregui/ODTW`.

5 Experimental Study

We explore the practical performance of the proposed ODTW measure by running several computer experiments aimed at two different yet related goals:

1. To show that ODTW is an efficient method and, hence, a suitable measure of dissimilarity, in terms of the computational complexity, for on-line classification scenarios.
2. To provide practical evidence of the capacity of the ODTW measure and its memory mechanism to react and accommodate concept changes in the processed streaming time series.

5.1 Efficiency

As for the first goal, we compare the running time of ODTW and conventional DTW methods. In this experiment two streaming time series have been produced by drawing one sample at a time from a uniform distribution with support [0,1] from an initial length of $n = m = 3$ samples to a maximum of $n = m = 70$ samples. For the sake of fairness, the same value of the Sakoe-Chiba band width $l = 50$ has been used to compute both DTW and ODTW dissimilarities. Under these modeling assumptions, the complexity of the DTW measure is expected to be quadratic, $\mathcal{O}(n^2)$, as long as the length of the time series is less than the band width, i.e, $n \leq l$. However, for $n > l$ the DTW complexity is $\mathcal{O}(ln)$. As addressed in Section 4, the computational complexity of the proposed ODTW is $\mathcal{O}(n)$ when $n \leq l$ and $\mathcal{O}(l)$ when $n > l$.

Figure 4 shows the running time (in seconds when implemented natively in Python 2.7 on a single i7 core at 3.10GHz) required to compute the ODTW (black) and DTW (gray) measures. A red dashed vertical line is included in the plot to indicate the value of the Sakoe-Chiba band width l. The empirical results shown in this plot support the hypothesis 1 discussed above, and show that ODTW is a suitable measure, in computational complexity terms, to quantify the dissimilarity between two streaming time series.

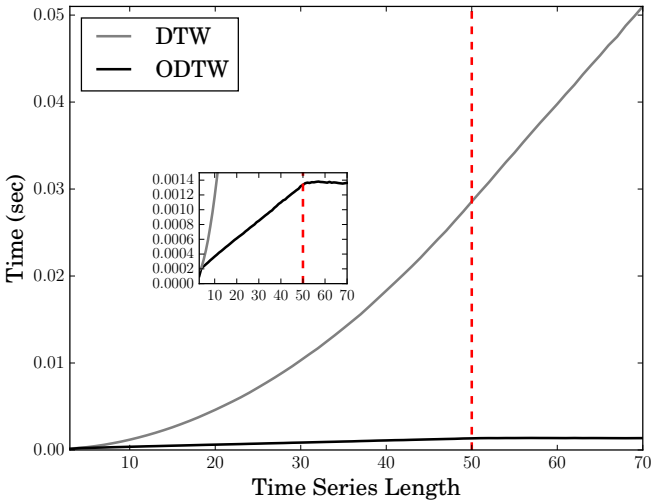


Fig. 4. Experimental running times required for the computation of the ODTW and DTW measures between two streaming time series of increasing length.

5.2 Predictive Performance

The second goal targeted in our simulation benchmark aims at assessing the predictive accuracy attained by a 1-NN classifier using the proposed ODTW metric when facing non-stationary streaming time series. To this end, we have monitored the evolution of the classifier accuracy for a given value of l and different values of parameter ρ. At this point, we recall that ρ allows balancing between the capability of the model to recover from drift concepts (low values of ρ) and its capacity to align warped time series and achieve better performance scores (high values of ρ). The experiment evaluates the so-called **prequential accuracy** $pACC(n)$ over different non-stationary datasets:

$$pACC(n) = \begin{cases} ACC_{sample}(n) & \text{if } n = n_{ref}, \\ pACC(n-1) + \frac{ACC_{sample}(n) - pACC(n-1)}{n - n_{ref} + 1} & \text{otherwise,} \end{cases} \quad (5)$$

where $ACC_{sample}(n)$ is 0 if the prediction of the sample at time n is wrong and 1 if correct; and n_{ref} denotes a time step that allows resetting the prequential accuracy at times where we force a drift to occur through the stream. This allows analyzing how the accuracy evolves after a drift, independently of the previous behavior of the classifier.

In order to analyze the prequential accuracy in a systematic way, synthetic datasets for on-line classification of stream time series have been designed and utilized for this second set of experiments. Such artificial datasets are built upon several publicly available time series datasets commonly used in DTW-based time series analysis, which can all be retrieved from the UCR Time Series Classification Archive [4]. In particular, we will use the datasets listed in Table 1.

Given different values of the weight parameter ρ, the designed datasets should allow analyzing the ability of the ODTW measure to adjust to time series stationarity. Therefore, streaming time series should have at least two different stationary parts and be periodic in each stationary interval. We generalize this intuitive premise to build the reference and query streams for each dataset in Table 1. In particular, query streams are composed by an endless repetition of stationary intervals (concepts), each formed by P time series of the same class drawn uniformly at random from the corresponding subset. In order to simulate a non-stationarity in the stream, the generation process avoids repeating the same class between every two consecutive stationary periods. Reference streams are composed for every label in the dataset by concatenating uniformly sampled time streams for every label in the dataset. Consequently, each query sequence in the dataset presents a recurrent concept change occurring every P time series. For instance, query class labels in a binary classification problem with stationary periods of $P = 3$ time series of length 2 samples each would be given by $\{0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 0, 0, \ldots\}$.

Table 1. Main characteristics of the utilized UCR datasets

Name	Train/TestSize	Time series length (n)	No. Classes	1-NN score	Sakoe-Chiba band l
`Gun Point`	50/150	150	2	0.913	0
`Italy Power Demand`	67/1029	24	2	0.955	0
`Two Lead ECG`	23/1139	82	2	0.868	4
`Face Four`	24/88	350	4	0.886	7
`CBF`	30/900	128	3	0.964	14
`Toe Segmentation 2`	36/130	343	2	0.908	17

We construct a total of 50 different test instances for each dataset in Table 1 to obtain an estimate of the average prequential accuracy achieved by an ODTW-based 1-NN classifier when predicting the label associated to the stream upon the arrival of every sample. Results are collected in Figure 5 for every dataset with its optimal value of l (Table 1) and different choices of the memory parameter ρ. Vertical dashed lines indicate the time at which the end of a time series meets the beginning of the next time series (periodicity of the stationary interval), being highlighted in bold black if the transition involves a label change. In these experiments, a class label change is produced every $P = 3$ time series. The horizontal dashed line indicates the best DTW-based 1-NN accuracy rate reported in [4] and listed in Table 1. The values taken by ρ have been chosen according to the length of the original time series. Particularly, the chosen values correspond to $\rho^m \in \{0.0001, 0.01, 0.1, 0.5, 1\}$. These parameters represent very-short-, short-, middle-, long- and full-range memory. The value of the band width l – designed to sacrifice the ODTW flexibility for a computationally efficient computation – has been set equal to the optimal Sakoe-Chiba band width found for every dataset shown in Table 1.

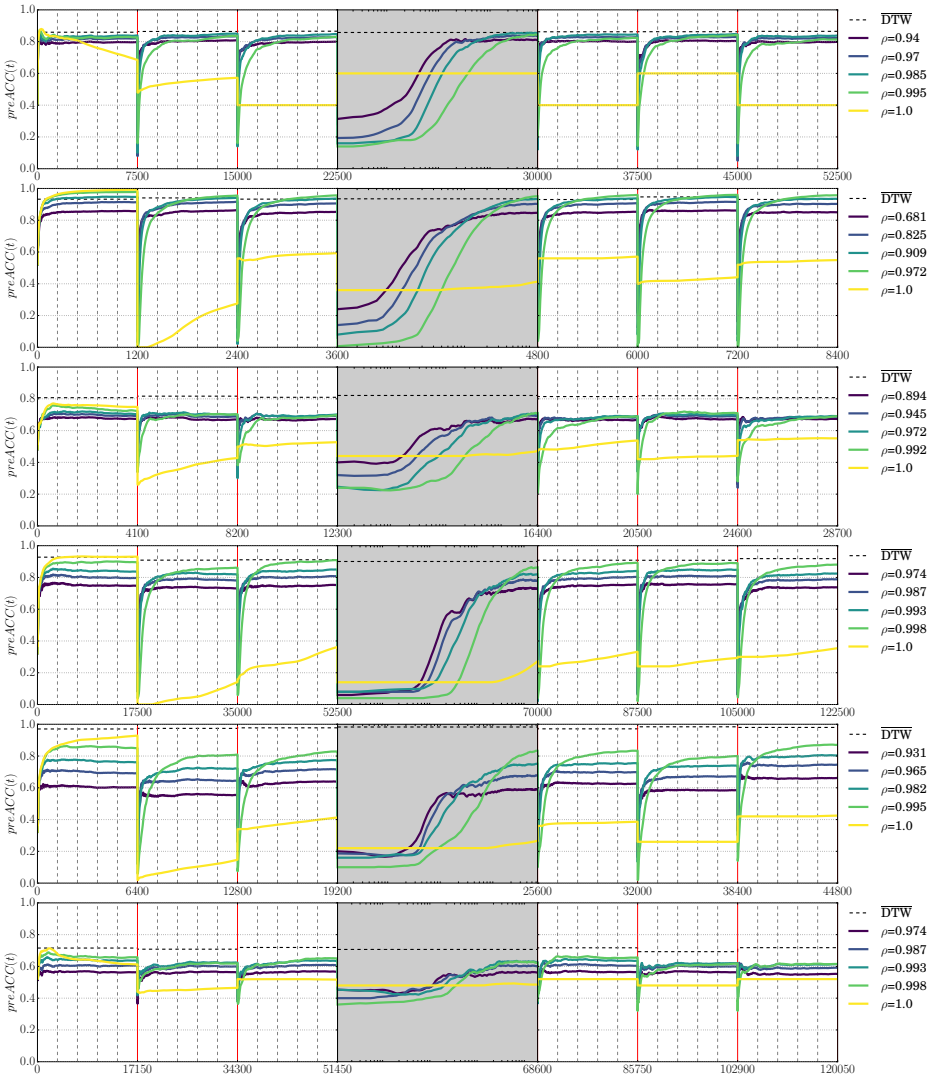


Fig. 5. Evolution of the prequential accuracy $pACC(n)$ over 7 stationary intervals of $P = 50$ concatenated time series each achieved by a 1-NN model using the proposed ODTW measure with the optimal l for each case and different values of ρ. The grey shadowed area corresponds to the 4-th stationary interval where the horizontal axis is represented in logarithmic scale. From top to bottom, plotted results correspond to `Gun point`, `Italy Power Demand`, `Two Lead ECG`, `Face Four`, `CBF` and `Toe Segmentation 2` datasets.

.

Several conclusions can be drawn from the experiments in Figure 5:

– As the value of ρ decreases (yielding a lower influence of past observations in the ODTW computation), the 1-NN model reacts more quickly to stationary changes, hence the prequential accuracy increases faster. However, in certain cases this involves a penalty in accuracy once the concept has become stable: the reason lies in the fact that the weighted memory of the ODTW measure fails to exploit past distance information of relevance for a discriminative alignment between the time series.
– As the values of ρ increase (a higher influence of past observations in the ODTW computation), even though the 1-NN model requires more time to recover from the stationary changes, its prequential accuracy tends to be better once the concept has been learned. In addition, as ρ increase the variance of the obtained prequential accuracy decreases. When $\rho = 1$, 1-NN performs worst due to the lack of a forgetting mechanism, which makes the value of the ODTW measure strongly biased by past alignments not linked to the concept to be predicted.

In light of the experimental results discussed above, we conclude that ODTW has both the reduced complexity and the flexibility to adapt to non-stationary environments needed for efficiently dealing with streaming time series.

6 Conclusions and Future Work

In this work, we have presented the On-line Dynamic Time Warping (ODTW), a natural adaptation of the popular DTW to the streaming time series setting. ODTW can be computed efficiently in an incremental way by avoiding unnecessary recalculations (see Section 4.1). It includes two parameters, l and ρ, that can be used in order to adapt the proposed measure to the particularities of the streaming time series under analysis. On the one hand, the band width parameter l is inspired by the Sakoe-Chiba band approach and can be used to control the trade-off between the complexity of the incremental computation of ODTW (linear in l) and the ability to shrink or stretch the time axis in order to align two time series (see Section 4.1). On the other hand, ρ is the forgetting parameter and it can be used to control the memory of ODTW by giving less importance to past values. By controlling the memory of the proposed measure, we can adjust the ability of ODTW to react to drift changes in streaming time series (see Section 4.2).

Due to the efficiency and flexibility of ODTW for dealing with streaming time series, we plan to extend its principles to other popular Elastic Similarity Measures such as the Edit distance, EDR and ERP. In addition, we will extend the experimentation by incorporating other on-line problems with streaming time series. Similar to DTW in off-line learning tasks, ODTW can also be used in on-line supervised and unsupervised problems such as on-line clustering, classification, outlier detection, etc. which is very useful in many different real world applications; for example, gesture classification, load profiling in energy grids and fraud detection.

Acknowledgments

This work has been supported by the Basque Government through the ELKARTEK program (ref. BID3ABI KK-2016/00096). Aritz Pérez is partially supported by the ELKARTEK program from the Basque Government, and by the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa excellence accreditation SVP-2014-068574 and SEV-2013-0323.

References

1. Cavalcante, R.C., Minku, L.L., Oliveira, A.L.: Fedd: Feature extraction for explicit concept drift detection in time series. In: Neural Networks (IJCNN), 2016 International Joint Conference on. pp. 740–747. IEEE (2016)
2. Chen, L., Ng, R.: On the marriage of lp-norms and edit distance. In: Proceedings of the Thirtieth international conference on Very large data bases-Volume 30. pp. 792–803. VLDB Endowment (2004)
3. Chen, L., Özsu, M.T., Oria, V.: Robust and fast similarity search for moving object trajectories. In: Proceedings of the 2005 ACM SIGMOD international conference on Management of data. pp. 491–502. ACM (2005)
4. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.: The ucr time series classification archive (July 2015), `www.cs.ucr.edu/~eamonn/time_series_data/`
5. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the VLDB Endowment 1(2), 1542–1552 (2008)
6. Ditzler, G., Roveri, M., Alippi, C., Polikar, R.: Learning in nonstationary environments: A survey. IEEE Computational Intelligence Magazine 10(4), 12–25 (2015)
7. Faisal, M.A., Aung, Z., Williams, J.R., Sanchez, A.: Data-stream-based intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study. IEEE Systems Journal 9(1), 31–44 (2015)
8. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing 23(1), 67–72 (1975)
9. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recognition 44(9), 2231–2240 (2011)
10. Keogh, E.: Exact indexing of dynamic time warping. In: Proceedings of the 28th international conference on Very Large Data Bases. pp. 406–417. VLDB Endowment (2002)
11. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 285–289. ACM (2000)
12. Keogh, E.J., Pazzani, M.J.: Derivative dynamic time warping. In: Proceedings of the 2001 SIAM International Conference on Data Mining. pp. 1–11. SIAM (2001)
13. Krawczyk, B., Minku, L.L., Gama, J., Stefanowski, J., Woźniak, M.: Ensemble learning for data stream analysis: A survey. Information Fusion 37, 132–156 (2017)
14. Lin, J., Keogh, E., Lonardi, S., Chiu, B.: A symbolic representation of time series, with implications for streaming algorithms. In: Proceedings of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery. pp. 2–11. ACM (2003)

15. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery 29(3), 565–592 (2015)
16. Rakthanmanon, T., Campana, B., Mueen, A., Batista, G., Westover, B., Zhu, Q., Zakaria, J., Keogh, E.: Searching and mining trillions of time series subsequences under dynamic time warping. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. pp. 262–270. ACM (2012)
17. Rath, T.M., Manmatha, R.: Word image matching using dynamic time warping. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on. vol. 2, pp. II–II. IEEE (2003)
18. Ristad, E.S., Yianilos, P.N.: Learning string-edit distance. IEEE Transactions on Pattern Analysis and Machine Intelligence 20(5), 522–532 (1998)
19. Rodrigues, P.P., Gama, J., Pedroso, J.: Hierarchical clustering of time-series data streams. IEEE transactions on knowledge and data engineering 20(5), 615–627 (2008)
20. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE transactions on acoustics, speech, and signal processing 26(1), 43–49 (1978)
21. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: Ftw: fast similarity search under the time warping distance. In: Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. pp. 326–337. ACM (2005)
22. Salvador, S., Chan, P.: Toward accurate dynamic time warping in linear time and space. Intelligent Data Analysis 11(5), 561–580 (2007)
23. Yu, D., Yu, X., Hu, Q., Liu, J., Wu, A.: Dynamic time warping constraint learning for large margin nearest neighbor classification. Information Sciences 181(13), 2787–2796 (2011)
24. Zhao, X., Li, X., Pang, C., Zhu, X., Sheng, Q.Z.: Online human gesture recognition from motion data streams. In: Proceedings of the 21st ACM international conference on Multimedia. pp. 23–32. ACM (2013)