Prefix and Suffix Invariant Dynamic Time Warping

Diego F. Silva, Gustavo E. A. P. A. Batista
Universidade de São Paulo
{diegofsilva, gbatista}@icmc.usp.br

Eamonn Keogh
University of California, Riverside
eamonn@cs.ucr.edu

Abstract **— While there exist a plethora of classification algorithms for most data types, there is an increasing acceptance that the unique properties of time series mean that the combination of nearest neighbor classifiers and Dynamic Time Warping (DTW) is very competitive across a host of domains, from medicine to astronomy to environmental sensors. While there has been significant progress in improving the efficiency and effectiveness of DTW in recent years, in this work we demonstrate that an underappreciated issue can significantly degrade the accuracy of DTW in real-world deployments. This issue has probably escaped the attention of the very active time series research community because of its reliance on static highly contrived benchmark datasets, rather than real world dynamic datasets where the problem tends to manifest itself. In essence, the issue is that DTW's eponymous invariance to warping is only true for the main "body" of the two time series being compared. However, for the "head" and "tail" of the time series, the DTW algorithm affords no warping invariance. The effect of this is that tiny differences at the beginning or end of the time series (which may be either consequential or simply the result of poor "cropping") will tend to contribute disproportionally to the estimated similarity, producing incorrect classifications. In this work, we show that this effect is real, and reduces the performance of the algorithm. We further show that we can fix the issue with a subtle redesign of the DTW algorithm, and that we can learn an appropriate setting for the extra parameter we introduced. We further demonstrate that our generalization is amiable to all the optimizations that make DTW tractable for large datasets.**

Keywords— Time Series, Dynamic Time Warping, Similarity

I. Introduction

Among all the time series mining tasks, query-by-content is the most basic. It is the fundamental subroutine used to support nearest-neighbor classification, clustering, etc. The last decade has seen mounting empirical evidence that the properties of time series mean that Dynamic Time Warping (DTW) is the best distance measure for across virtually all domains [20].

However, virtually all current research efforts assume a perfect segmentation of the time series. This assumption is engendered by the availability of dozens of contrived datasets from the UCR time series archive [4]. Improvements on this (admittedly very useful) resource have been seen as sufficient to warrant publication of a new idea, but it would be better to see success on these benchmarks as being only necessary to warrant consideration of a new approach.

In particular, the way in which the majority of the datasets were created and "cleaned" means that algorithms that do well on these datasets can still fail when applied to real world streaming data.

The issue lends itself to a visually intuitive explanation. Fig. 1 shows two examples from the Australian Sign Language dataset aligned by DTW. We can see the utility of DTW here, as it aligns the later peak of the blue (bold) time series to the earlier occurring peak in the red (fine) time series. However, this figure also illustrates a weakness of DTW. Because every point must be matched, the first few points in the red sequence are forced to match the first point in the blue sequence.

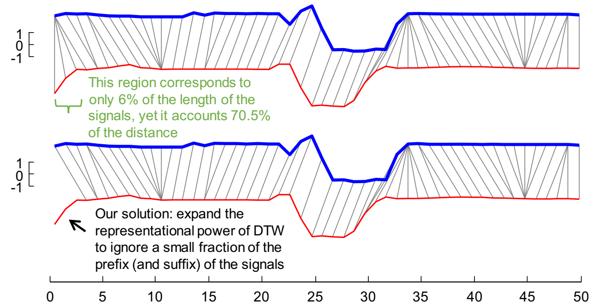

Fig. 1. *top*) Two time series compared with DTW. While the prefix of the **red** (fine) time series consists of only 6% of the length, it is responsible for 70.5% of the distance. *bottom*) We propose to address this disproportionate appointing of error by selectively ignoring parts of the prefix (and/or suffix)

While Fig. 1 does show the problem on a real data object, the reader may wonder how common this issue is "in the wild". We claim that at least in some domains, this problem is very common.

For example, heartbeat extraction algorithms often segment the signal to begin at the maximum of the QRS complex [16]. However, this location has the greatest variability in its prefixes and suffixes. Likewise, star light curves, for which DTW is known to be very effective, have cycles extracted by a technique called universal phasing [13]. However, universal phasing has the unfortunate side effect of placing the maximum variance at the prefix and suffix of the signals.

In this work, we address this problem of uninformative and undesirable "information" contained just before and just after the temporal measurement of informative data. For the sake of clarity, we will refer to these unwanted values as *prefix* and *suffix*, and use *endpoints* to refer to both.

Our approach is simple and intuitive, but highly effective. We modify the endpoint constraint of Dynamic Time Warping (DTW) to provide endpoint invariance. The main idea behind our proposal is allowing DTW to ignore some leading/trailing values in one or both of the two time series under comparison. While our idea is simple, it must be carefully executed. It is clear that ignoring too much (useful) data is just as undesirable as paying attention to spurious data.

2374-8486/16 $31.00 © 2016 IEEE
DOI 10.1109/ICDM.2016.107

We note that somewhat similar observations were known to the signal processing community when DTW was the state-of-the-art technique for speech processing (in the 1980's and 90's before being superseded by Markov models [10]). However, the importance of endpoint invariance for time series seems to be largely unknown or underappreciated [8][11][12].

We can summarize the main contributions of this paper as follows:

- We draw the data mining community's attention to the endpoint invariance, which seems to be a little or no considered issue;
- We propose a modification of the well-known algorithm Dynamic Time Warping to provide invariance to endpoints;
- Although simple and intuitive, we show that our method can considerably improve the classification accuracy when warranted, and just as importantly, our ideas do not reduce classification accuracy if the dataset happens to not need endpoint invariance;
- Unlike other potential fixes, our distance measure respects the property of symmetry and, consequently, can be applied in a multitude of data mining algorithms with no pathological errors caused by the order of the data;
- In spite of the fact that we must add a parameter to DTW, we show that it is possible to robustly learn a good value for this parameter using only the training data.

II. Time Series Suffix and Prefix

Most research efforts for time series classification assume that all the time series in the training and test sets are carefully segmented by using the precise endpoints of the desirable event [12][13][18][20]. Despite the ubiquity of time series datasets that fulfill such an assumption, in practical situations the exact endpoints of events are difficult to detect. In general, a perfectly segmented dataset can only be achieved by manual segmentation or some contrivance that uses external information.

To see this, we revisit the Gun-Point dataset, which has been used in more than two hundred papers to test the accuracy of time series classification [4]. The data objects considered in such a set have perfectly flat prefixes and suffixes. However, these were obtained only by carefully prompting the actor's movements (pointing a gun or a finger) with a metronome that produced an audible cue every five seconds.

In more realistic scenarios, the event of pointing a gun/finger must be detected among several different movements. Before drawing the weapon, the actor could be running, talking on a cell phone, etc.

For example, consider the scenario in which some movement was performed just before the weapon was aimed. In addition, another movement started immediately after the gun was returned to the holster. In this case, the time series could have a more complex shape as shown in Fig. 2. As visually explained in Fig. 1, it is clear that prefix and suffix would greatly prejudice the distance estimation in this case.

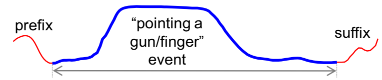

Fig. 2. Example of a time series containing the event to be classified (in blue) and prefix and suffix information (in red)

Another possible issue that can result from automatic segmentation is the algorithm used to extract the time series be too "aggressive" and make the mistake of truncating the last few observations of the event of interest. Obviously, a similar issue could also happen at the *beginning* of the signal.

In this case, the time series is missing its true suffix. Even with such missing information, the shape that describes the beginning of the action *may* be enough such that it will be classified correctly. However, the object that would otherwise be considered its nearest neighbor may contain information of the entire movement, as shown in Fig. 2. To classify this kind of badly cropped item correctly, a distance measure must avoid matching the last few observations of the complete event to the values observed in our badly segmented event. In Section V we will show how our method can solve these issues.

III. Definitions and Background

The Dynamic Time Warping (DTW) is arguably the most useful distance measure for time series analysis. For example, mounting empirical evidence strongly suggest that the simple nearest neighbor algorithm using DTW outperforms more "sophisticated" time series classification methods in a wide range of application domains [20].

In contrast to other distance measures, such as those in the Lp-norm family, the DTW computes a non-linear alignment between the observations of the two time series being compared. DTW computes such an optimal alignment under endpoint, monotonicity and continuity constraints [12]. The main focus of this work is the endpoint constraint, defined as the following.

Endpoint constraint. The matching between a pair of time series x and y, with lengths n and m respectively, starts at the pair of observations $(1,1)$ and ends at (n,m).

An additional constraint commonly applied to DTW is the warping constraint. The most common warping constraint for DTW is the Sakoe-Chiba warping window [14], which limits the time difference that the algorithm is allowed to match the observations. The benefit of using a warping constraint is two fold: the DTW calculation takes less time (as it is not necessary to calculate values for the entire cost matrix) and it avoids pathological alignments. As a practical confirmation of its utility using the constraint, we note that it has been shown to improve classification accuracy [12].

For a complete description of the algorithm for calculating the DTW, we refer the reader to [15].

IV. Related Work

The time series mining method that shares more similarities to our proposal is the open-end DTW (OE-DTW) [17]. However, OE-DTW was proposed to match incomplete time series to complete references. In other words, such a method is based on the assumption that we can construct a training set with carefully cropped time series and we can know the exact point that represents the beginning of the time series to be classified.

Specifically, OE-DTW is a method that allows ignoring any amount of points at the end of the training time series, i.e., the alignment may finish in any observation of the reference object.

A weakness of the OE-DTW is that it does not consider the existence of prefix information. A modification of the OE-DTW called open-begin-end DTW (OBE-DTW) or subsequence DTW [9] mitigates this issue. OBE-DTW allows the match of observations to start at any position of the training time series.

Although OBE-DTW recognizes that both prefix and suffix issues may exist, it only addresses the problem in the training time series. A more important observation is that OBE-DTW is not symmetric, which severely affects its utility. For example, the clustering results obtained by OBE-DTW are dependent on the order in which the algorithm processes the data.

In addition to this issue, OBE-DTW has one other fatal flaw. In essence, it can be "too invariant," potentially inducing meaningless alignments in some cases, causing loss of useful information in the time series.

Similar to the OBE-DTW, the method proposed in this paper is based on a relaxation of the endpoint constraint. However, our method is symmetric and strictly limits the amount of the signals that can be ignored, preventing the meaningless alignments.

V. PREFIX AND SUFFIX-INVARIANT DTW (Ψ-DTW)

While there are many different methods proposed for time series classification (decision trees, etc.), it is known that the simple nearest neighbor is extremely competitive in a wide range of applications and conditions [20]. Given this, the only decision left to the user is the choice of the distance measure. In most cases, this choice is guided by the invariances required by the task and domain [3]. In conjunction with simple techniques, such as z-normalization, DTW can provide several invariances like amplitude, offset and the warping (or local scaling) itself.

In this work, we address what we feel is the "missing invariance," the invariance to spurious prefix and suffix information. Given the nature of our proposal, we call our method Prefix and Suffix-Invariant DTW, or simply PSI-DTW (or ψ-DTW).

This paper proposes a relaxed version of the endpoint constraint, defined as the following.

Relaxed endpoint constraint. Given an integer value r, the alignment path between the time series x and y starts at any pair of observations in $\{(1,c_1+1)\} \cup \{(c_1+1,1)\}$ and ends at any pair in $\{(n-c_2,m)\} \cup \{(n,m-c_2)\}$, such that c_1 and $c_2 \in [0,r]$.

This relaxed constraint can avoid undesirable matches at the endpoints of any x or y time series by removing the obligation for the alignment path to start and end in the first and last pairs of observations.

The value r used in this definition is the relaxation factor parameter that needs to be defined by the user. We recognize the general undesirability of adding a new parameter to an algorithm. However, we argue it is necessary. In addition, we show that we are able to learn an appropriate r solely from the training data. We will return to this topic in Section VI.

An important aspect of the proposed endpoint constraint is the fact that, by definition, the same number of points is "relaxed" for both time series under comparison. This is what guarantees the symmetry of ψ-DTW.

The relaxation of endpoints slightly affects the initialization of the traditional DTW. Equation 1 defines the initialization of ψ-DTW.

$$\mathrm{dtw}(i,j)= \begin{cases} \infty, \text{ if } (i=0 \text{ and } j>r) \text{ or } (j=0 \text{ and } i>r) \\ 0, \text{ if } (i=0 \text{ and } j\leq r) \text{ or } (j=0 \text{ and } i\leq r) \end{cases} \quad (1)$$

In order to find the optimal non-linear alignment between the time series x and y, ψ-DTW follows the same recurrence relation than the classical DTW, defined by Equation 2.

$$\mathrm{dtw}(i,j)= \mathrm{c}\left(x_i,y_j\right)+\min \begin{cases} \mathrm{dtw}(i\text{-}1,j) \\ \mathrm{dtw}(i,j\text{-}1) \\ \mathrm{dtw}(i\text{-}1,j\text{-}1) \end{cases} \quad (2)$$

where $i \in [1,n]$ and $j \in [1,m]$, m being the length of the time series y. The partial $c(x_i,y_j)$ represents the cost of matching two observations x_i and y_j and is calculated by the squared Euclidean distance between them. Finally, the ultimate distance estimate can be directly obtained by the definition of the proposed relaxed endpoint constraint. Formally, the final distance calculation is given by Equation 3.

$$\begin{gathered} \Psi\text{-DTW}(x,y,r)= \min_{(i,j)\in \mathrm{finalSet}} [\mathrm{dtw}(i,j)], \\ \mathrm{finalSet} = \{(n\text{-}c_1,m)\} \cup \{(n,m\text{-}c_2)\} \;\forall\; c_1,c_2 \in [0,r]. \end{gathered} \quad (3)$$

For concreteness, Algorithm I describes ψ-DTW in detail.

ALGORITHM I. Ψ-DTW IMPLEMENTATION

Procedure ψ-DTW(*x,y,r*)
Input: Two user provided time series, ***x*** and ***y*** and the relaxation factor parameter ***r***
Output: The ψ-DTW distance between ***x*** and ***y***

1	$n \leftarrow \mathrm{length}(x)$, $m \leftarrow \mathrm{length}(y)$
2	$M \leftarrow$ infinity_matrix($n+1$,$m+1$)
3	$M([0,r],0) \leftarrow 0$
4	$M(0,[0,r]) \leftarrow 0$
5	**for** $i \leftarrow 1$ **to** n
6	**for** $j \leftarrow 1$ **to** m
7	$M(i,j) \leftarrow c(x_i,y_j) + \min(M(i\text{-}1,j\text{-}1), M(i,j\text{-}1), M(i\text{-}1,j))$
8	minX $\leftarrow \min(M([n\text{-}r,n],m))$, minY $\leftarrow \min(M(n,[m\text{-}r,m]))$
9	**return** min(*minX*,*minY*)

The algorithm starts by defining the variables used to access the length of time series (line 1) and the DTW matrix according to Equation 1 (lines 2 to 4). The *for* loops (lines 5 to 7) fill the matrix according to the recurrence relation defined in Equation 2. Finally, the algorithm finds the minimum value in the region defined by the new endpoint constrained and returns it as the distance estimate (lines 8 and 9). To implement the constrained warping version of this algorithm, one only needs to modify the interval of the second for loop (line 6) according to the constraint definition.

Note that the proposed method is a generalization of DTW, thus it is possible to obtain the classic DTW by our method. Specifically, if $r=0$, the final result of our algorithm is exactly the same as the classic DTW.

VI. Experimental Evaluation

This section summarizes the results obtained in our experimental evaluation. Because we are committed to reproducibility, we have made available all the source code, datasets, detailed results and additional experiments in a companion website for this work[1]. In addition to reproducing our experiments, the interested reader can use our code on their own datasets. We implemented all our ideas in Matlab, as it is ubiquitous in the data mining community.

To test the robustness of our method, we compare its performance against the accuracy obtained by the classic DTW, in both unconstrained- and constrained-warping versions (c.f. Section III). We refer to the constrained versions of the algorithms with names containing the letter c. For clarity, cDTW refers to the DTW with warping constraint and ψ-cDTW stands for the constrained version of ψ-DTW. In addition, we present results obtained using OBE-DTW.

We are not directly interested in studying the effect of warping window width on classification accuracy. The value of the warping window width parameter has been shown to greatly affect accuracy, but it has also been shown to be easy to learn a good setting for this parameter with cross-validation [12][18][20]. For simplicity, we fixed it as 10% of the length of the query time series by default.

However, this setting limits the choice of the relaxation factor to ψ-DTW. For any relaxation factor that is greater than or equal to the warping length, the final distance estimate does not change. It happens because the "open" cells outside the region defined by warping window are ignored by the algorithm. For this reason, when we wanted to test the effect of larger relaxation factors, the warping window used in the experiment was set by the same value as r.

In this experimental evaluation, we apply ψ-DTW on real datasets extracted in a scenario in which we do not have perfect knowledge or control over the events' endpoints. In some cases, the original datasets were obtained by recording sessions, in which the invariance to endpoints is enforced by the data collection procedure. In this case, we model the real world conditions by ignoring the external cues or annotations. In particular, we simulated a randomly-ordered stream of events followed by a classic subsequence extraction step. For this phase, we considered the simple sliding window approach. For additional details on the extraction phase, please refer to the companion website.

In keeping with common practice, we adopted the use of dictionaries as training data. A data dictionary is a subset of the original training set containing only its most relevant examples. The utility of creating dictionaries is two-fold [6]: it makes the classifier faster and the accuracy obtained by dictionaries is typically better than that obtained by using all the training data, which may contain outliers or mislabeled data. To compute the relevance of training examples to the classification task, we used the SimpleRank function [18].

The length of subsequences and the size of the dictionary for each dataset were chosen in order to obtain the best accuracy in the training set by using cDTW. In addition, the SimpleRank was also implemented using the classic cDTW instead of the ψ-DTW or the ψ-cDTW. This was done to ensure we are not biasing our experimental analysis in favor of our method.

Once created the dictionary, we need to estimate a good value for the parameter r. For this, we experimented with a wide range of possible values. We set r as a relative value to the length of the time series. Specifically, we used a set of values $rl_r \in \{0, 0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$, such that $r = \lceil n * rl_r \rceil$, where n is the length of the time series. We limited the value of r to be at most half the number of observations of the time series in order to avoid meaningless alignments.

Note that the choice of the size of the dictionary is a crucial determinant of the time complexity of the algorithm. In order to keep the algorithm fast, the number of objects in the dictionary tends to be small, which makes learning r difficult if we use the dictionary exclusively. In order to learn the value of r, we used a validation set containing all the training time series but those chosen as part of the dictionary. However, we notice that cross-validation techniques on the training set lead to similar results.

In the next sections, we describe the datasets used as case studies, followed by the summarized results. We notice again that the reader can find the detailed results in the companion website for this work.

1) Motor Current Data

Our first case study considers electric motor current signals [5]. The data in question includes 21 classes representing different motor operating conditions. In addition to a class that represents a diversity of healthy operation, the other classes represent defects in the apparatus (in particular, one to ten broken bars and one to ten broken end-ring connectors).

The original data used in this study is segmented, but with no attention paid to avoiding endpoints inconsistences. Therefore, instead simulating a data stream, we segmented the original time series using a static window placed in the middle of each time series. With this procedure, the signals have different endpoints in each different length we consider.

2) Robot Surface and Activity Identification

In this case study, we consider the classification of signals collected by the accelerometer embedded in a Sony ERS-210 Aibo Robot [19]. This robot is a dog-like model equipped with a tri-axial accelerometer to record its movements.

Using the streaming data sets collected by this robot, we evaluated the classification accuracy in two different scenarios: surface and activity recognition. In the former scenario, the goal is to identify the type of surface in which the robot is walking on. Specifically, the target classes for this problem are carpet, field, and cement.

[1] http://sites.google.com/site/psidtw/

In the second scenario, the aim is the identification of the activity performed by the robot. In this case, the target classes are the robot playing soccer, standing in a stationary position, trying to walk with one leg hooked, and walking straight into a fixed wall.

3) Gesture Recognition

Gesture recognition is one of the most studied tasks in the time series classification literature. The automatic identification of human gestures has become an increasingly popular mode of human-computer interaction.

In this study, we used the Palm Graffiti Digits dataset [1], which consists of recordings of different subjects "drawing" digits in the air while facing a 3D camera. The goal of this task is the classification of the digits drawn by the subjects.

4) Sign Language Recognition

Another specific scenario with gesture data used in this work is the recognition of sign language. A sign language is an alternative way to communicate by gestures and body language that replace (or augment) the acoustic communication. In this work, we used a dataset of Australian Sign Language (AUSLAN) [7]. The original dataset is composed of signs separately recorded in different sections. We used 10 arbitrarily chosen signs of each recording session displaced as a data stream.

5) Human Activity Recognition

Due to the growth in the use of mobile devices containing movement sensors (such as accelerometers and gyroscopes), there is also a notable increase in the interest of human activity analyses using this kind of equipment.

In this final case study, we investigate the robustness of ψ-DTW in the recognition of human activities using smartphone accelerometers. For this purpose, we used the dataset that first appeared in [2]. Originally, the recordings are composed of 128 observations of three coordinates of the device's accelerometers. In our study, we used the x-coordinate disposed in a streaming fashion.

6) Summary of the Results and Hypothesis Test

The results show that ψ-DTW achieves better results than the classical DTW and the OBE-DTW in most of the experimented cases. To make this observation stronger, a hypothesis test on the accuracies obtained by both methods was used. For this, we performed a paired Wilcoxon signed-rank test for comparing the performances of each combination between DTW, cDTW, and OBE-DTW against ψ-DTW and ψ-cDTW. Using a confidence factor of 95%, the test rejected the null hypothesis (that the mean ranks are similar) for all the comparisons.

It is interesting to analyze the accuracies obtained for each dataset, considering the best time series length. To evaluate this, we used the validation procedure applied to learn r as the method to choose the time series length to assess the performance of ψ-DTW. For comparison, we used the best accuracy obtained by OBE-DTW and DTW. Note that this analysis is favoring the competing algorithms, given that we used an *oracle* instead of learning the best series length for these methods. TABLE I. shows the result of this experiment.

TABLE I. ACCURACIES OBTAINED BY OBE-DTW, DTW, AND Ψ-DTW

Dataset	OBE-DTW	DTW	ψ-DTW	cDTW	ψ-cDTW
AUSLAN	0.503	0.500	0.579	0.490	0.514
Human Activity	0.555	0.558	0.575	0.566	0.578
Motor Current	0.114	0.119	0.400	0.119	0.405
Palm Graffiti	0.262	0.374	0.355	0.363	0.363
Robot Activity	0.839	0.845	0.854	0.822	0.833
Robot Surface	0.950	0.846	0.910	0.842	0.842

VII. LOWER BOUNDING OF Ψ-DTW

One of the biggest concerns while designing a new distance measure is time efficiency. This is more prevalent in our case since we are proposing a modification of Dynamic Time Warping, an $O(n^2)$ algorithm. In fact, a straightforward implementation of the nearest neighbor algorithm under DTW makes its use impractical on large datasets. For this reason, the community has proposed several methods to improve the efficiency of the similarity search under DTW.

Specifically, [11] shows that the combination of few simple techniques for speeding-up similarity search makes possible to handle truly massive data under DTW. We claim that all these methods can be applied to the ψ-DTW with subtle or no modifications.

Some of the most important speed-up methods rely on the use of a lower bound (LB) function. An LB function returns a value certainly lower or equal to the true DTW between two objects. Consider that we have a variable *best-so-far* that stores the distance to the nearest neighbor know up to the current iteration of the search algorithm. For each time series in the training set, we first calculate the LB between it and the query. Clearly, if the LB function returns a value greater than the *best-so-far*, the training object is not the nearest neighbor of the query. Therefore, the current object can be discarded before having its distance to the query estimated. We can extend this to a k-nearest neighbor scenario by simply replacing the *best-so-far* by the distance to the k-th nearest object known at that moment.

Now we are in position to answer the question "How can we use previously proposed LB functions with ψ-DTW?". Adapting an LB function to ψ-DTW requires the analysis of the possible first and last observation pairs. For sake of exemplification, we will adopt the most widely used LB function, the LB_Keogh [8].

The calculation of LB_Keogh consists of two main steps. The first step is the estimation of an envelope to a given query time series q of length n. Specifically, the envelope is composed of an upper sequence $U=(U_1,U_2,\ldots,U_n)$ and a lower sequence $L=(L_1,L_2,\ldots,L_n)$. Fig. 3 exemplifies the upper and lower sequences of a given query time series.

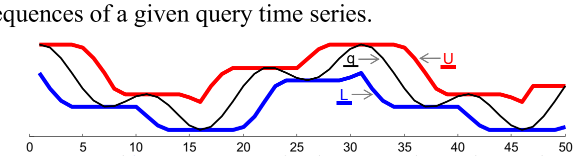

Fig. 3. Upper and lower sequences of a given query time series q estimated by LB_Keogh

Once the envelope is calculated, we estimate the value of the LB function. For a time series t to be compared to the query q, the LB_Keogh is calculated as the Euclidean distance between the observations of t that falls outside the envelope and the nearest upper or lower sequence.

In order to adapt LB_Keogh to our method, we need to relax its endpoints. Since ψ-DTW can skip the matching of the first and last r observations in either q or t, the LB function should ignore these values. We call the adapted LB function ψ-LB_Keogh, and define it formally in Equation 4.

$$\psi\text{-LB\_Keogh}(q,t)=\sum_{i=r+1}^{n-r}\begin{cases}(t_i-U_i)^2, \text{ if } t_i>U_i\\(L_i-t_i)^2, \text{ if } t_i<L_i\\0 \text{ otherwise}\end{cases} \quad (4)$$

Fig. 4 illustrates the ψ-LB_Keogh between q and t.

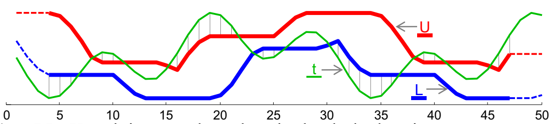

Fig. 4. ψ-LB_Keogh ignores the values in the dashed regions

The pruning power of a lower bound function is directly related to its tightness, i.e., its value needs to be close to the true DTW. To demonstrate the tightness of ψ-LB_Keogh, we compared it with the tightness of LB_Keogh. We quantified the tightness of the LBs by dividing them by the corresponding DTW distances, setting the warping window as 10% of the time series length. The relaxation factor takes the same value. TABLE II. shows the results obtained in the training set with the shortest time series used in each study case.

TABLE II. TIGHTNESS OF LB_KEOGH AND Ψ-LB_KEOGH

Dataset	LB_Keogh	ψ-LB_Keogh
AUSLAN	0.522	0.484
Human Activity	0.173	0.152
Motor Current	0.259	0.292
Palm Graffiti Digits	0.549	0.490
Sony Robot Activity	0.120	0.110
Sony Robot Surface	0.174	0.151

From these results, we can note that the tightness of both methods is similar. This indicates that the endpoint constraint relaxation does not impair the tightness of ψ-LB_Keogh.

VIII. CONCLUSION

In this paper, we proposed a modification of the endpoint constraint of DTW to make it suffix- and prefix-invariant. In addition to be simple and intuitive, our method is quite effective. Experimental results show that our method outperforms the classic DTW by a large margin in datasets that contain spurious endpoints. In addition, we demonstrated that the distance obtained by our method can be tightly lower bounded by a slight modification of the current lower bounds of DTW, which indicates that our modified DTW is tractable for large datasets.

For the sake of clarity and brevity, in this work we only discussed the application of our algorithm to classification. However, it can also be applied to a large variety of tasks, such as clustering, motif discovery, outlier detection, etc. We leave those explorations, including discussions on how to set the parameter r for each task, as future work.

REFERENCES

[1] J. Alon, V. Athitsos, Q. Yuan, and S. Sclaroff, "A unified framework for gesture recognition and spatiotemporal gesture segmentation". *IEEE T. Pattern. Anal*. vol. 31, no. 9, pp. 1685-1699, 2009.

[2] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine", in *Proceedings of the International Workshop of Ambient Assisted Living*, 2012, pp. 216-223.

[3] G. E. A. P. A. Batista, E. Keogh, O. M. Tataw, and V. M. A. Souza, "CID: an efficient complexity-invariant distance for time series", *Data Min. Knowl. Discov.*, vol. 28, no. 3, pp. 634-669, 2014.

[4] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. E. A. P. A. Batista, "The UCR Time Series Classification Archive", 2015, URL www.cs.ucr.edu/~eamonn/time_series_data/

[5] N. A. Demerdash, and J. F. Bangura, "Characterization of induction motors in adjustable-speed drives using a time-stepping coupled finite-element state-space method including experimental validation", *IEEE T. Ind. Appl.*, vol. 35, no. 4, pp. 790-802, 1999.

[6] B. Hu, Y. Chen, and E. Keogh, "Time Series Classification under More Realistic Assumptions", in *Proceedings of the SIAM International Conference on Data Mining*, 2013, pp. 578-586.

[7] M. W. Kadous, "Temporal Classification: Extending the Classification Paradigm to Multivariate Time Series", Doctoral Thesis, University of New South Wales, 2002.

[8] E. Keogh, and C. A. Ratanamahatana, "Exact indexing of dynamic time warping", *Knowl. Inf. Syst.*, vol. 7, no. 3, pp. 358-386, 2005.

[9] M. Müller, "Dynamic Time Warping". *Information retrieval for music and motion*. Springer, pp. 69-84, 2007.

[10] L. R. Rabiner, and B. Juang, B., "Pattern-comparison techniques", *Fundamentals of speech recognition,* Prentice-Hall, Inc., pp. 141-241, 1993.

[11] T. Rakthanmanon, B. Campana, A. Mueen, G. E. A. P. A. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. "Searching and mining trillions of time series subsequences under dynamic time warping", in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2012, pp. 262-270.

[12] C. A. Ratanamahatana, and E. Keogh, "Three Myths about Dynamic Time Warping Data Mining", in *Proceedings of the SIAM International Conference on Data Mining*, 2005, pp. 506-510.

[13] U. Rebbapragada, P. Protopapas, C. Brodley, and C. Alcock, "Finding anomalous periodic time series", *Mach. Learn., vol.* 74, no. 3, pp. 281-313, 2009.

[14] H. Sakoe, and S. Chiba, "Dynamic Programming Algorithm Optimization for Spoken Word Recognition", *IEEE T. Acoust. Speech*, vol. 26, no. 1, pp. 43-49, 1978.

[15] D. F. Silva, and G. E. A. P. A. Batista, "Speeding Up All-Pairwise Dynamic Time Warping Matrix Calculation", in *Proceedings of the SIAM International Conference on Data Mining*, 2015, pp. 837-845.

[16] A. Suhrbier, R. Heringer, T. Walther, H. Malberg, and N. Wessel, "Comparison of three methods for beat-to-beat-interval extraction from continuous blood pressure and electrocardiogram with respect to heart rate variability analysis", *Biomed. Tech.*, vol. 51, on. 2, pp. 70-76, 2006.

[17] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli, "Matching incomplete time series with dynamic time warping: an algorithm and an application to post-stroke rehabilitation", *Artif. Intell. Med.*, vol. 45, no. 1, pp. 11-34, 2009.

[18] K. Ueno, X. Xi, E. Keogh, and D.-J. Lee, "Anytime Classification Using the Nearest Neighbor Algorithm with Applications to Stream Mining", in *Proceedings of the IEEE International Conference on Data Mining*, 2006, pp. 623-632.

[19] D. Vail, and M. Veloso, "Learning from accelerometer data on a legged robot", in *Proceedings of the IFAC/EURON Symposium on Intelligent Autonomous Vehicles*, 2004.

[20] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data", *Data Min. Knowl. Discov.*, vol. 26, no. 2, pp. 275-309, 2013.