

Generalized Dynamic Time Warping: Unleashing the Warping Power Hidden in Point-Wise Distances

Rodica Neamtu ^{#1}, Ramoza Ahsan ^{#2}, Elke A. Rundensteiner ^{#3}, Gabor Sarkozy ^{#4},
Eamonn Keogh ^{&5}, Hoang Anh Dau ^{&6} Cuong Nguyen ^{#7} Charles Lovering ^{#8}

[#] Worcester Polytechnic Institute, Worcester MA, USA

& UC Riverside, Riverside CA, USA

¹rneamtu@wpi.edu, ²rashan@wpi.edu, ³rundenst@wpi.edu, ⁴gsarkozy@wpi.edu,
⁵eamonn@ucr.edu, ⁶hdau001@ucr.edu, ⁷ctnguyendinh@wpi.edu, ⁸cjlovering@wpi.edu

Abstract—Domain-specific distances preferred by analysts for exploring similarities among time series tend to be “point-to-point” distances. Unfortunately, this point-wise nature limits their ability to perform meaningful comparisons between sequences of different lengths and with temporal mis-alignments. Analysts instead need “elastic” alignment tools such as Dynamic Time Warping (DTW) to perform such flexible comparisons. However, the existing alignment tools are limited in that they do not incorporate diverse distances. To address this shortcoming, our work introduces the first conceptual framework called Generalized Dynamic Time Warping (GDTW) that supports now alignment (warping) of a large array of domain-specific distances in a uniform manner. While the classic DTW and its prior extensions focus on the Euclidean Distance, our GDTW is the first method that generalizes the ubiquitous DTW and “extends” its warping capabilities to a rich diversity of point-to-point distances. Based on our GDTW paradigm that preserves the efficiency of the dynamic programming paradigm of DTW, we design an abstraction that implemented by our GDTW Design Tool enables analysts to “warp” new distances with little programming effort. Through extensive evaluation studies on 85 real public domain benchmark datasets, we show that our newly warped distances offer higher classification accuracy than the previously available distances for the majority of these datasets. Further, our case study on heart arrhythmia data illustrates the utility of the new distances enabled by our GDTW warping methodology.

I. INTRODUCTION

A. Motivation and Background

In our current era time series are produced at an unprecedented rate in many application domains in the form of stock fluctuations, electrocardiograms, and many others. Mining of these time series data sets relies on finding relationships based on distances between subsequences both within and across time series. Clearly, there is a broad range of important problems from retrieval, classification, and clustering to outlier detection that require such distance-based comparison capabilities. Yet, the selection of a distance¹ deployed for these data mining tasks strongly impacts the results. This can lead

¹From the mathematical point of view, the smaller the similarity distance (i.e. Euclidean) between two time series, the more similar they are. In contrast, the higher the value of the similarity measure (i.e Cosine) the more similar they are considered to be [1]. Similarity measures can be expressed in terms of distances. In this paper for simplicity, we will not distinguish between these two categories and refer to them simply as “distances”.

to poor or even erroneous results because the efficacy of a distance depends on the semantics and how relevant they are for the specific application.

In this vein, it has been well established that one cannot rely on one single distance; but instead application domains need a variety of distances to solve their specific problems [1]. For example, compound classification in chemistry selects the most relevant chemical descriptors using Minkowski distance [2], motion detection applications indexing d-dimensional trajectories prefer Chebyshev polynomials [3], while image retrieval reflecting human visual perception tends to utilize Manhattan [4], [5] and Mahalanobis distances [6].

Unfortunately, all distances mentioned above are only “point-to-point distances”. That is, they cannot be used for comparing sequences of different lengths or that are not aligned in time. Yet in many real-life scenarios, sequences tend to be of *different lengths*. For example, while some patients may be in the intensive care unit for just one afternoon, others may stay there hooked up to instruments generating medical signals for days or weeks.

Similarly, *misalignment of sequences* often occurs, meaning that certain patterns (shapes) may be stretched or compressed into distinct time periods in two time series. For example, the stock fluctuation of sales volumes by two competitors such as Google and Apple may never arise at the same time point (i.e., within the first year of their respective public offering). Or, they may not be stretched over the same time period (while a peak might be reached by Apple more rapidly with the release of a new device, a similar high might have been experienced by Google - but just distorted slower in time). Or a doctor might want to explore if certain shapes found in the ECG of a patient during a medical episode are also found in signals previously recorded for other patients with the same condition because this can help diagnosis.

Performing time series mining relies on using a chosen distance to compare sequences with misalignments, different lengths, or both. To address this requirement of more powerful matching, the notion of “elastic” distances has been introduced [7], [8], [9], [10]. For example, the peaks of the ECG shapes in Fig. 1 are misaligned in time. Elastic alignment tools are

needed for meaningful comparisons between these sequences. Such ECGs could help doctors diagnose arrhythmia related issues, referring to changes of the normal sequence of electrical impulses which may cause the heart to beat too fast, too slowly, or erratically. When the heart does not pump blood effectively, the lungs, brain and other organs cannot work properly and may shut down or be damaged. The red (solid)

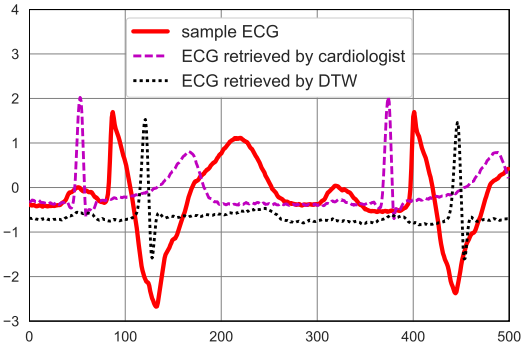


Fig. 1: ECG best match retrieved by doctor is different than the one retrieved by the classic DTW.

line in Fig. 1 displays the ECG of a target patient. Finding the best match for such sample sequence can help doctors identify specific heart conditions. In this scenario, based on her expertise, the cardiologist selected the purple (line-dash) sequence as best match to this target patient from the ECG collection. The black (dot-dash) sequence was retrieved as the best match to the sample by using the well-known DTW. As the figure shows, the doctor retrieved a different sequence than the one retrieved by DTW. Thus, in this scenario, DTW, despite its popularity, is not the most suitable distance for similarity search.

In fact, it has been shown that DTW [8] can be rigid due to incorporating the Euclidean as base distance [11], leading to unintuitive and even erroneous results at times. Because of the temporal misalignments the best match sequence could not have been retrieved by using a point-wise distance either. We postulate that this sequence could be retrieved by using a different robust alignment tool, one that does not incorporate the Euclidean as base distance. This hypothesis is confirmed by our arrhythmia related study (Sec. VII) where the sequence marked in black matching the choice of the cardiologist is indeed retrieved by a newly warped distance designed using our GDTW methodology.

As we describe below, no methodology exists that enables analysts to work with the point-to-point distance most appropriate for their domain, yet empower this targeted distance to support robust time-warped matching. Application developers thus face the dilemma of having to choose between the most appropriate distance to tackle their problem or a distance supporting flexible sequence matching. There is thus a need to enable analysts to use the distance best suited for their specific domain without limiting them to only compare sequences of the same length or without any local shifting. In this work we now provide the first solution to this open problem, that

is we design a method that uniformly extends warping to a wide diversity of distances, regardless of their mathematical expressions.

B. State-of-the-Art and Its Limitations

Elastic distances including Dynamic Time Warping [7], [8], Longest Common Subsequence (LCSS) [9], and Edit Distance with Real Penalty (EDR) [10] enable elastic sequence matching. Particularly, Dynamic Time Warping (DTW) [8], popular for time series data mining, allows sequences to be stretched or compressed along the time axis, i.e., a point of one sequence can be matched to one or more points of another sequence. DTW has become increasingly popular due to its expressiveness – being applied to RNA expression data in bioinformatics [12], ECG pattern matching in medicine [13], and aligning biometric data in surveillance systems [14].

Despite its popularity, DTW has been shown to not always be the most appropriate distance for exploring time series because it can produce pathological results through non-intuitive alignments [11]. One reason for this shortcoming stems from the fact that DTW is restricted to using the Euclidean distance as its base distance [8]. This limits its utility for applications that require other distances as we further describe.

As we show in Sec. VIII, countless modifications of the classic DTW have been proposed to either (1) optimize its performance by indexing, caching, and other system optimizations [9], [15], [7], [16], [17] or (2) improve the quality of alignments between time series [11], [18], [19]. Closer to our work is [20], which uses sum-based distances in the dynamic programming strategy. However, this method [20] only incorporates simple metrics based on sums such as Euclidean or Manhattan – falling short in handling many of the point-to-point distances motivated above.

In short, none of the state-of-art DTW warping methods supports any of the popular distances, such as Minkowski, Chebyshev, Sorensen, Cosine, Pearson, Jaccard, etc. These distances are based on combinations of mathematical operations such as fractions, products, min, and max. We illustrate in this paper that such distances can now all be successfully “warped” using our proposed framework.

C. Our Proposed GDTW Framework

In this work, we overcome the problem of extending warping abilities to diverse point-wise distances by designing a universal alignment tool, called Generalized Dynamic Time Warping. GDTW is flexible enough to use different point-to-point distances in computing the warping path, yet powerful enough to enable time warping. Defining such generalized distance is not enough, it must be complemented by efficient strategies for being computed over large time series datasets.

Our work fundamentally changes the classic DTW, while keeping its main purpose of robustness to local misalignments in time. We propose a step-by-step methodology that enables a large number of point-wise distances to perform warping and do so efficiently. Our new GDTW framework now enables the community to develop new warped distances with ease. The

resulting GDTW-empowered time-alignment tools can then be leveraged for a broad range of problems from classification, clustering, to best match retrieval.

Contributions.

- We introduce the first conceptual framework, called GDTW, whose core mathematical foundation enables analysts to transform a diversity of point-wise distances into elastic distances by extending warping properties using a uniform approach (Sec. III-A and III-B).
- We devise a multi-step methodology that empowers analysts to “warp” their desired point-wise distance and efficiently compute the generalized warping path using a general dynamic programming strategy (Sec. III-C and III-D).
- To realize the formal GDTW framework, we model the key principles of GDTW-based warped distance specifications by three core abstractions that allow analysts to define new distances, construct their recursive expressions and incorporate them into the DP strategy. The resulting API implemented by the template-based GDTW Design Tool enables analysts to “warp” new distances without much programming effort (Sec. IV).
- We validate our GDTW framework theoretically and practically by applying it to popular point-wise distances with diverse mathematical characteristics [1], including distances that could not work under the classic DTW algorithm, e.g., Minkowski and Sorensen. This results in a repository of warped distances – a valuable resource for the community (Sec. V).
- Our extensive experimental study on the 85 datasets from diverse application domains from the benchmark UCR Archive [21] shows the effectiveness of our newly warped distances for a variety of data mining tasks compared to the state-of-the-art DTW (Sec. VI-B, VI-C, VI-D).
- Our study of Arrhythmia data guided by domain experts shows the utility of GDTW distances for better interpreting ECG similarity in the medical domain (Sec. VII).

II. CLASSIC DYNAMIC TIME WARPING

Suppose we have two time series $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_m)$. To align these sequences using DTW, an $n \times m$ matrix $M(X, Y)$ is constructed, where the $(i, j)^{th}$ element of the matrix is the Euclidean Distance between x_i and y_j , i.e., $w_{i,j} = ED(x_i, y_j)$. Then a *warping path* P is a set of elements that forms a path in the matrix from $(1, 1)$ to (n, m) . The t^{th} element of P denoted as $p_t = (i_t, j_t)$ refers to the indices i_t, j_t of (x_{i_t}, y_{j_t}) of this matrix element in the path. Thus a path P is $P = (p_1, p_2, \dots, p_t, \dots, p_T)$, where $n \leq T \leq 2n - 1$, $p_1 = (1, 1)$ and $p_T = (n, m)$.

Definition 1: Warping Path Weight: Given two time series $X = (x_1, \dots, x_n)$ and $Y = (y_1, \dots, y_m)$, the weight of the warping path P is defined as:

$$w(P) = \sqrt{\sum_{t=1}^T w_{i_t, j_t}^2}. \quad (1)$$

The DTW distance then is defined to be the weight of the path with the minimum weight ($\min_P(w(P))$).

A warping path is subject to the following constraints:

1. Boundary condition. $p_1 = (1, 1)$ and $p_T = (n, m)$ or the path has to start and end on the opposite corners of the matrix.
2. Continuity condition. The steps in the warping path are restricted to adjacent cells, including diagonally adjacent cells. Using simplified notations [11], for $p_i = (u, v)$ we have $p_{i-1} = (u', v')$, where $u - u' \leq 1$ and $v - v' \leq 1$.
3. Monotonicity condition. The elements on the path must monotonically progress in one direction, namely $u - u' \geq 0$ and $v - v' \geq 0$ and $(u, v) \neq (u', v')$.

Since there is an exponential number of warping paths satisfying these conditions, finding the minimum weight warping path is prohibitively expensive. Fortunately, the warping path can be efficiently calculated by using dynamic programming [20]. Conceptually, given the matrix M containing pairwise Euclidean distances of all elements in the sequences X and Y , we construct a dynamic programming matrix Γ by filling in the values using the following recursive expression.

$$\gamma(i, j) = ED^2(x_i, y_j) + \min(\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1))$$

The current distance $\gamma(i, j)$ in the *cell* (i, j) is computed as the sum of the square of the distance currently found in the cell in the same position in the original matrix M and the minimum of the cumulative distances found in the adjacent cells (diagonal, left and down) in the dynamic programming matrix Γ . Then $DTW^2(X, Y) = \gamma(n, m)$. Further details can be found in [22] and [17].

Fig. 2 illustrates an example of computing the classic DTW warping path for two sequences X and Y as depicted with values in bold font. The leftmost matrix M from the classic DTW algorithm contains the pairwise square ED between the elements of the sequences. The middle matrix Γ showcases the dynamic programming strategy used for computing the path. For example, as indicated on the red dotted arrow, the element 0 in matrix M is summed with the minimum of the three elements (left, down and diagonally-down) in Γ leading to the value 0 in Γ . The gray values indicate that values are calculated “as needed” to find the path efficiently. Lastly, the matrix on the right highlights the resulting path in blue.

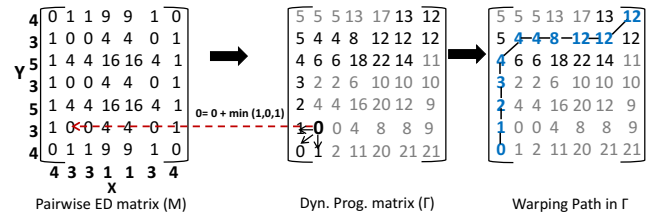


Fig. 2: Computing the warping path with classic DTW

III. GENERALIZED TIME WARPING

A. Towards a Generalized Distance

We design the GDTW framework to preserve all advantages of DTW, while also supporting the transformation of a wide array of popular point-to-point distances d into their warped counterparts $GDTW_d$. Better yet, unlike previous work, our GDTW approach “empowers” analysts to warp existing point-to-point distances d of their choice. We offer efficient strategies for computing these warping paths for distances meeting the recursive and symmetry properties described below. Our approach fundamentally changes the algorithm for computing the weight of the warping path by generalizing it to allow the embedding of alternate distances, regardless of their mathematical expressions. This overcomes the limitations of previous approaches which at best can only “warp” distances based on simple sums.

While our generalized DTW can be applied conceptually to any point-to-point distance, it is important in practice to compute it efficiently. Thus, we change the DP strategy from the classical DTW algorithm and adapt it to work in our generalized context.

B. Fundamentals of GDTW Warping Path

We define the concept of a general warping path and explain how to incorporate new functions in computing it. Given two sequences $X = (x_1, x_2, \dots, x_n)$ and $Y = (y_1, y_2, \dots, y_m)$, with $n \geq m$, we construct an $n \times m$ grid graph G , as a generalization of the matrix Γ from the classic DTW. As shown in Fig. 3, we define a *warping path* P as a sequence of elements that forms a contiguous path from $(1, 1)$ to (n, m) . By “decoding” this general warping path and extracting the values for x_{i_k} and y_{j_k} at every position on the path, we conceptually construct the following two equal-length vectors: $X_P = (x_{i_1}, x_{i_2}, \dots, x_{i_T})$ and $Y_P = (y_{j_1}, y_{j_2}, \dots, y_{j_T})$, where some of the x_{i_k} and y_{j_k} are repeated while advancing on the path. Considering an arbitrary distance d , the weight of the warping path P is then defined as the distance between X_P and Y_P , which is computed using d . That is, we have $w(P) = d(X_P, Y_P)$.

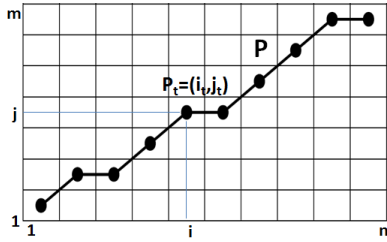


Fig. 3: General Warping Path

Definition 2: The **Generalized Dynamic Time Warping Distance** corresponding to a distance d , denoted by $GDTW_d$, is the weight of the path P with the minimum weight, namely:

$$GDTW_d(X, Y) = \min_P (d(X_P, Y_P)).$$

Theoretically the generalized dynamic time warping distance, as in Def. 2, can incorporate any distance, not just based on

sums, but on maximum, minimum, fractions of sums, products, etc. However, as written, it requires us to find all warping paths first, then determine their weight, and lastly pick the one path with the minimum weight. This is not feasible in practice.

Thus, the key idea is that we must be able to construct the distance function recursively by indicating how to incorporate the n^{th} coordinates in the distance measure based on the previous $n-1$ coordinates. For this, we introduce a key recursive property that must first be identified and then utilized to define and compute the weight of the warping paths.

Definition 3: The distance measure d in Def. 2 must satisfy the *recursive condition* below. There exists a 3-variable function $f_d : \mathbf{R}^+ \times \mathbf{R} \times \mathbf{R} \rightarrow \mathbf{R}^+$ where \mathbf{R} denotes the set of real numbers and \mathbf{R}^+ denotes the set of non-negative real numbers with respect to a distance d such that for vectors $X_P = (x_1, x_2, \dots, x_n)$ and $Y_P = (y_1, y_2, \dots, y_n)$ ($n \geq 2$), we have:

$$\begin{aligned} d(X_P, Y_P) &= d((x_1, \dots, x_n), (y_1, \dots, y_n)) = \\ &= f_d(d((x_1, \dots, x_{n-1}), (y_1, \dots, y_{n-1})), x_n, y_n). \end{aligned} \quad (2)$$

The f_d function tells us, given the distance measure on the first $n-1$ coordinates $(x_1, \dots, x_{n-1}, y_1, \dots, y_{n-1})$, how to incorporate the n^{th} coordinates (x_n, y_n) . This expression assumes that the distance measure is symmetric in the coordinates. This means swapping the order of any two coordinates (x_i, x_j) and (y_i, y_j) in both their respective sequences does not change the distance between these sequences.

As we show in Sec. V-A, with some mathematical effort all distances in [1] are good candidates for being warped using our methodology because they follow the two requirements for a distance d to be efficiently warped, as follows:

1. The distance measure d is symmetric in the coordinates. This means swapping the order of any two coordinates (x_i, x_j) and (y_i, y_j) in both their respective sequences does not change the distance between these sequences.
2. The distance d can be written in a recursive manner as per the recursive condition defined above.

To illustrate the $GDTW$ with a concrete example, we now re-examine the well-known Euclidean Distance (ED) [23], previously applied in the classic DTW, in our new context. That is, we give the recurrence for ED as per Def. 3.

Euclidean Distance Example: Given the Euclidean distance (ED) between two sequences X and Y defined as

$$d_{ED}(X, Y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}. \quad (3)$$

The **recursive expression** of ED according to Def. 3 is:

$$f_{d_{ED}}(a, x_n, y_n) = \sqrt{a^2 + |x_n - y_n|^2}, \quad (4)$$

where a is the value of the function d_{ED} for the first $n-1$ coordinates.

C. Efficient Strategy for Computing the GDTW Warping Path

Let us assume that we have a distance measure d satisfying Eqn (2). Then we propose to *change* the DP strategy designed for the classical DTW (see Sec. II) and adapt it to work with any distance, regardless of its mathematical expression. The classic DP strategy always computes the SUM between the previous value and the min of the three cumulative values of the base distance advancing on the path, therefor it can only work if the base distance is a sum. The new DP strategy instead computes the MIN of the three values calculated as the base distance between the previous value and each of the cumulative values respectively. This new strategy incorporates *the core mathematical operation of each base distance*, i.e. max for Minkowski or Chebyshev, fraction for Sorensen, etc., while the classic strategy was simply based on sum.

Definition 4: The dynamic programming **general recursive expression** for warping a distance d is:

$$\gamma(i, j) = \min \begin{cases} f_d(\gamma(i-1, j-1), x_i, y_j), \\ f_d(\gamma(i-1, j), x_i, y_j), \\ f_d(\gamma(i, j-1), x_i, y_j). \end{cases} \quad (5)$$

with $\gamma(1, 1) = d(x_1, y_1)$.

Definition 5: Using Eqn (5), the “warped” version of a distance d returns a **general dynamic warping distance** defined as:

$$GDTW_d(X, Y) = \gamma(n, m) \quad (6)$$

Given a distance d , we first design the function f_d in Eqn (2). Thereafter, we plug the former into Eqn (5) to derive a DP solution that computes the warped version of distance d .

To illustrate, we apply the above process to our running example of Euclidean distance. Namely, by replacing $f_{d_{ED}}$ from Eqn (4) in Eqn (5), we derive the following dynamic programming recurrence for warping the Euclidean distance:

$$\gamma(i, j) = \min \begin{cases} (\gamma(i-1, j-1)^2 + |x_i - y_j|^2)^{\frac{1}{2}}, \\ (\gamma(i-1, j)^2 + |x_i - y_j|^2)^{\frac{1}{2}}, \\ (\gamma(i, j-1)^2 + |x_i - y_j|^2)^{\frac{1}{2}}. \end{cases}$$

We note that the DP recursive expressions derived from (5), when applied to ED, are identical to those in the classic DTW.

D. Proposed GDTW Methodology

In brief, the formal steps of our GDTW methodology for creating a corresponding warping distance $GDTW_d$ for a given distance d are:

- 1) *Select* a desired distance d as the potential warping candidate. If the distance d satisfies the recursive condition (Sec. III-B), then the following steps provide the strategy to efficiently compute the warping path²
- 2) *Design* the function f_d for the recursive expression in Def. 2 to serve as the weight of the corresponding GDTW warping path (Sec. III-B). This step is crucial to efficiently computing the warping path.

²If the distance d is not already in our repository and it does not satisfy this condition, other strategies can be devised to compute the warping path.

- 3) *Find* the recursive dynamic programming expression by plugging f_d into Eqn (5) to efficiently compute the weight of the path (Sec. III-C).

IV. GDTW DESIGN TOOL FOR NEW DISTANCES

Once the appropriate expressions have been designed following the specifications in Sec. III-D we built an API to support the incorporation of new warped distances by analysts with minimal programming effort. The use of templates to abstract the distance formulas provides a flexible interface, while avoiding the overhead of dynamic dispatch that would be incurred by polymorphic operations.

As first step, auxiliary variables are defined to serve as containers to store intermediate values in the calculation of a distance. The auxiliary variables effectively hold the accumulated values required by the distance recursive function, i.e. the $f_d((x_1..x_{n-1}), (y_1..y_{n-1}))$ in Def. 3. For simple distances there is only one such variable. For example for ED this variable accumulates the squared difference of pairs of data points. More complex distances may require more than one auxiliary variable to remain incrementally computable. As we show later (Sec. V), the Sorensen distance needs two variables: one to accumulate the absolute difference and the other the absolute sum of two data points. Lastly, we calculate the final value of the Sorensen distance by dividing the first variable by the second one. A simple auxiliary variable can be set as a floating-point variable. Multiple variables can be collectively defined as C++ structures or simply elements in an array.

We define our new distance as a class with the following set of methods (with A being the type of auxiliary variables):

- A `init()` **initializes** the auxiliary variables.
- A `reduce(A prev, data_t Xi, data_t Yi)` combines the auxiliary variables in `prev` and the data points `Xi` and `Yi` from the two time series (`data_t` represents the data type of each point) according to the recursive expression of f_d in Def. 3.
- `data_t finalize(A raw_final, TS X, TS Y)` calculates the final numeric value from the auxiliary variables.

We note that although the distance classes do not extend any common base class, the existence and signature of these methods are enforced by the compiler via the structure of the template. Therefore, we benefit from error detection during compile time. Given the newly defined distance metric class and the type of auxiliary variables, the compiler generates a point-wise distance function and a warped distance function, both embedded within the `init()`, `reduce()`, and `finalize()` methods in the specific distance metric. The structures of function variants are uniform across distances.

The key point is that we offer a user-friendly API to define new warped distances with ease, yet without sacrificing execution performance costs. That is, our generalized system does not impose any significant overhead in the similarity matching time. This is achieved due to our well-engineered templated solution using meta-programming with C++ templates.

V. COMMUNITY RESOURCE OF WARPED DISTANCES

A. Designing Warped Distances Using GDTW Methodology

We now show how our proposed GDTW framework can be used to warp diverse distances using our GDTW methodology (Sec. III-D). We focus on distances collected in the highly cited survey paper [1] due to its large coverage of popular point-to-point distances. In particular, we showcase well-known distances such as Manhattan, Minkowski and Sorensen, popular in studying similarity of time series. Minkowski (same mathematical expression as Chebyshev) is based on max and thus the classic DTW algorithm valid only for sum-based distances would fail. Similarly, based on a fraction of sums, Sorensen could not work using the classic DTW either.

Our study achieves three objectives:

- (1) It demonstrates the utility of the GDTW methodology for warping in a consistent manner a rich diversity of distances composed of complex expressions including division, square root, max, min, fractions and products.
 - (2) Our work constructs a valuable “start-up” repository of off-the-shelf warped important distances ready to use by anyone.
 - (3) The availability of these examples will help designers of distances in the future find the needed recurrence expressions.
- L_p -distances in general, for $p=1$ and $p=2$, leading to Manhattan and respectively Euclidean distances: Given the L_p distance between two time series X and Y defined as:

$$d_{L_p}(X, Y) = \left(\sum_{i=1}^n |x_i - y_i|^p \right)^{\frac{1}{p}}, \quad (7)$$

the recursive expression of the L_p distance is stated as:

$$f_{d_{L_p}}(a, x_n, y_n) = (a^p + |x_n - y_n|^p)^{\frac{1}{p}},$$

where a is the total value of the distance measured for the first $n-1$ coordinates. The p in the L_p can be plugged in accordingly to model specific L_p norms, as mentioned above. The dynamic programming recurrence for warping L_p distances is:

$$\gamma(i, j) = \min \begin{cases} (\gamma(i-1, j-1))^p + |x_i - y_j|^p)^{\frac{1}{p}}, \\ (\gamma(i-1, j))^p + |x_i - y_j|^p)^{\frac{1}{p}}, \\ (\gamma(i, j-1))^p + |x_i - y_j|^p)^{\frac{1}{p}}. \end{cases}$$

Euclidean Distance was reviewed earlier in Sec. III-B (and thus not repeated here), so we show now Manhattan distance.

Manhattan Distance: Given the Manhattan distance d_{MD} between two time series X and Y, defined as:

$$d_{MD}(X, Y) = \sum_{i=1}^n |x_i - y_i|, \quad (8)$$

its recursive expression is:

$$f_{d_{MD}}(a, x_n, y_n) = (a + |x_n - y_n|) \quad (9)$$

where a is the value of $f_{d_{MD}}$ for the first $n-1$ coordinates. The recursive dynamic programming is:

$$\gamma(i, j) = \min \begin{cases} (\gamma(i-1, j-1) + |x_i - y_j|), \\ (\gamma(i-1, j) + |x_i - y_j|), \\ (\gamma(i, j-1) + |x_i - y_j|). \end{cases}$$

Similarly to Fig. 2, we give an example for computing the warping path for the same pair of sequences using $GDTW_{MD}$ in Fig. 4. We note here that the resulting path differs from the path found by the classic DTW.

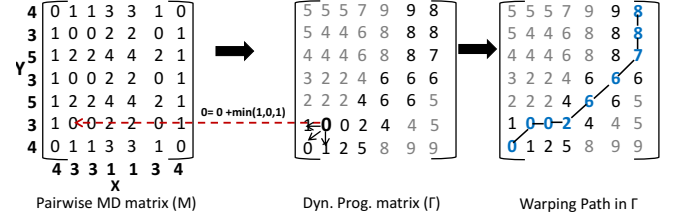


Fig. 4: Computing the warping path with $GDTW_{MD}$.

Minkowski Distance: Given the Minkowski distance d_{Mink} between two time series X and Y defined as:

$$d_{Mink}(X, Y) = \max_{i=1}^n |x_i - y_i|, \quad (10)$$

its recursive expression is:

$$f_{d_{Mink}}(a, x_n, y_n) = \max(a, |x_n - y_n|). \quad (11)$$

where a is the value of $f_{d_{Mink}}$ for the first $n-1$ coordinates. The dynamic programming recursive expression is:

$$\gamma(i, j) = \min \begin{cases} \max(\gamma(i-1, j-1), |x_i - y_j|), \\ \max(\gamma(i-1, j), |x_i - y_j|), \\ \max(\gamma(i, j-1), |x_i - y_j|). \end{cases}$$

Similarly to Fig. 2, we give an example for computing

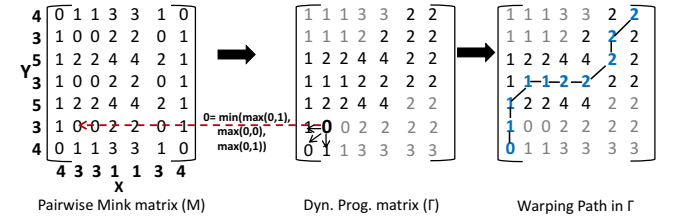


Fig. 5: Computing the warping path with $GDTW_{Mink}$.

the warping path for the same pair of sequences using $GDTW_{Mink}$ in Fig. 5. We note two differences: the resulting path is different than all previous paths, and there is no sum in the new DP strategy. Unlike DTW, this DP uses a completely different expression based on max not sum. The warped Minkowski distance could not work using the DP expression of the classic DTW.

Sorensen Distance: Sorensen, used in ecology [24], is another example that could not be accommodated by the classic DTW because of its complex form (a fraction of sums). Given Sorensen distance d_{sor} between X and Y as:

$$d_{sor}(X, Y) = \frac{\sum_{i=1}^n |x_i - y_i|}{\sum_{i=1}^n |x_i + y_i|} \quad (12)$$

Its recursive expression is:

$$f_{d_{Sor}}\left(\frac{a}{b}, x_n, y_n\right) = \frac{a + |x_n - y_n|}{b + |x_n + y_n|} = \frac{a'}{b'}. \quad (13)$$

where a and b denote the total value of the differences and respectively the sums of the first $n-1$ coordinates.

The dynamic programming recursive expression is:

$$\gamma(i, j) = \min \left\{ \begin{array}{l} \frac{a_1 + |x_i - y_j|}{b_1 + |x_i + y_j|}, \\ \frac{a_2 + |x_i - y_j|}{b_2 + |x_i + y_j|}, \\ \frac{a_3 + |x_i - y_j|}{b_3 + |x_i + y_j|}, \end{array} \right.$$

where $\gamma(i-1, j-1) = \frac{a_1}{b_1}$, $\gamma(i-1, j) = \frac{a_2}{b_2}$, $\gamma(i, j-1) = \frac{a_3}{b_3}$.

Warping Other Distances. Above illustrates that our new DP strategy is general enough to work with all distances in [1]. The key to warping these distances is to formulate a recursive expression to fit Def. 3 and then embed that into Eqn (5). Due to space constraints, we offer other examples, namely warping the Cosine distance in our extended repository of warped distances [25]. The Cosine distance, which measures the angles between two vectors, corresponds to the normalized Inner Product and also has a complex form that could not work with the DP of the classic DTW. Many other popular distances such as Jaccard, Dice and Pearson, based on similar arithmetic expressions can be warped with using our methodology.

Discussion of GDTW complexity. The complexity of DTW has been shown to be $O(mn)$ [8], where m and n are the lengths of the compared sequences. Although the recursive formulation for GDTW is different, it leads to the same complexity as DTW because it maintains the DP traversal strategy. The construction of the computation matrix dominates the complexity of GDTW. Each cell of the matrix is computed by comparing three values of the previous cells at every step, a constant cost as the three previous cells are already computed. The complexity of GDTW does not change with the base distance because of the simple point-wise nature of the distances computed at each step. Thus the traversal of the mn cells leads to an overall complexity of $O(mn)$.

B. Incorporating New Distances Using GDTW Design Tool

Next, we explain the integration of our API with our previous examples of “theoretically” warping these distances. In particular, we show the use of our Design Tool on two examples: warped Minkowski ($GDTW_{Mink}$) and warped Sorensen ($GDTW_{Sor}$). Classic DTW and $GDTW_{MD}$ both use sum. Thus their implementation is very similar to $GDTW_{Mink}$ and can be safely omitted here.

Implementing $GDTW_{Mink}$. Given that only one cumulative value, namely a , is used in the recursive expression of the distance in Eqn (11), we only need one auxiliary variable. This variable a is expressed as `prev` in the `reduce` method.

```
class Minkowski {
  data_t init() { return 0; }
  data_t reduce(data_t prev, data_t xi,
               data_t yi) {
    return max(prev, |xi-yi|); }
}
```

```
data_t finalize(data_t raw_final, TS X,
               TS Y) {
  return raw_final; }
}
```

Implementing $GDTW_{Sor}$. The Sorensen distance defined in Eqn (12) requires the use of two auxiliary variables. That is, the right side of Eqn (12) contains a fraction of two separate values, namely cumulative sum of the differences in coordinates denoted as `prev[0]` and cumulative sum of sums of coordinates denoted as `prev[1]` corresponding to the values a and b respectively in Eqn (13). The `reduce` method computes the warped distance according to the recursive expression in Eqn (13) by dividing the two values as shown below.

```
class Sorensen {
  data_t* init() { return new data_t[2]; }
  data_t* reduce(data_t* prev, data_t xi,
                data_t yi) {
    return {prev[0] + |xi-yi|,
            prev[1] + |xi+yi|}; }
  data_t finalize(data_t raw_final, TS X,
                 TS Y) {
    return raw_final[0]/raw_final[1]; }
}
```

To summarize, using our tool, we construct the warping variant $GDTW_d$ of the point-wise distance d by using the `init()`, `reduce()`, and `finalize()` methods. Our DP expression remains the same for all distances - regardless of their respective mathematical expressions, as shown above. The analyst simply has to design f_d to define `reduce()`, while the remaining work for “warping” d is done automatically by the system. We note that the recursive expressions for different distances will lead to differences in the empirical response times as we show in Sec. VI-C and VI-D. This is due to minor differences in the definitions of the distance, e.g., the cost for squaring is slightly different than the cost of performing an absolute value, etc.

VI. EXPERIMENTAL EVALUATION

A. Experimental Methodology

Our GDTW framework can be utilized to warp a plethora of distances as demonstrated by our study above. While it is well accepted in the literature (Sec. I-C) that different distances are preferred depending on the application domain, data set and mining task, GDTW, by now providing additional distances to analysts, would enrich their repertoire and with it their opportunity to find additional insights missed by current distances. In this light, we conduct a few studies to demonstrate that in some cases even simple new GDTW distances can already consistently beat the classic DTW. For this, we focus on a select subset of GDTW variants, namely, $GDTW_{Mink}$ (warped Minkowski or Chebyshev), $GDTW_{ED}$ (DTW), and $GDTW_{MD}$ (warped Manhattan). The reason for choosing these is three-fold: (1) they are well known to the research community, (2) we documented in the introduction that their point-wise versions are valuable in diverse domains, yet cannot perform flexible sequence matchings, (3) $GDTW_{Mink}$ was chosen specifically because it is based on max and it cannot work using the classic algorithm for DTW, but it now works

under our GDTW framework. Additional experimental results and detailed instructions to reproduce our experiments are found at [25].

Data Sets. We use the largest public collection focused on time-series datasets that we are aware of, the UCR Archive [21] containing 85 benchmark datasets from various domains.

Three Classes of Evaluation:

Experiment 1: Time Series Classification. We evaluate the effectiveness of our newly warped distances for time series classification. For this we apply each distance over the training and test sets of the 85 datasets in the UCR archive, using them as (parameter-free) 1-NN classifiers. We compute the classification accuracy, i.e., number of correctly classified instances over all instances, and the error rate in performing 1-NN classification. Because the 1-NN classifier is deterministic, we only perform this computation once.

Experiment 2: Best Match Retrieval and Clustering. We find the best match (or nearest neighbor) for a given sample query sequence first by using a “point-to-point” distance and then by its “warped” counterpart. We then compare these matches. Our experiment aims to show that: (1) the warped distances tend to return different results compared to their point-wise versions, as expected when the sequences are not aligned in time, and (2) diverse warped distances each may provide different insights (results) that would be missed by the other warped distances. In addition, we demonstrate the impact of using these new distances on an average linkage hierarchical clustering problem.

Experiment 3: Evaluation of Warping Characteristics. Similarly to the well-known Derivative Dynamic Time Warping method [11], which studies the “over-warping” produced by the classic DTW, we compute the *amount of warpings* produced by our GDTW variants for pairs of sequences as:

$$W = (l - \text{average}(m, n)) / \text{average}(m, n) \quad (14)$$

where $0 \leq W \leq 1$ and m, n are the lengths of the compared sequences. $W=0$ if the algorithm does not find a warping between two sequences. W increases to a maximum value of 1 as the warping “discovered” by the algorithm increases. Analysts interested in finding similar sequences with fewer warpings can utilize these findings. Similarly to [11], we also measure the *sensitivity of our warped distances to local distortion* by introducing distortions in a controlled fashion into pairs of synthetic sequences.

B. Classification Using 85 Diverse Time Series Benchmarks

Time series classification [10][26][27] frequently uses a distance as a subroutine in the K-Nearest Neighbor (K-NN) algorithm. This simple algorithm has been shown to be surprisingly competitive, by consistently outperforming rival methods such as decision trees, neural networks, Bayesian networks, and Hidden Markov Models [27], [28]. Moreover, time series classification has thus far been one of the few tasks to resist significant progress from “deep learning” [29]. Given this, the choice of “which” distance to use is important. Literally dozens of distances have been proposed (see [9], [29] and the references therein). However, an extensive recent

empirical comparison (performing 36 million experiments) has confirmed the excellent performance of classic DTW-based 1-NN (in our case $GDTW_{ED}$), which is only beaten by Ensemble Classifiers [29].

We now evaluate if other warped distances such as $GDTW_{MD}$ or $GDTW_{Mink}$ are even more effective for classifying time series. To test this research question, we performed classification experiments on all 85 datasets from the UCR Archive. The train/test splits were identical. We fixed the warping window to 100% for all experiments. Thus any differences can be attributed solely to the effect of changing the base distance. The raw results and the details of the experiments including pairwise comparisons using error-rate binary plots are archived at [25].

For brevity, we present here a compact visual summary displaying a comparison between the results obtained using the classic DTW (our $GDTW_{ED}$), $GDTW_{MD}$ and $GDTW_{Mink}$. Pairwise comparisons of distance measures are often presented as 2D-scatter-plots [26]. We compare three algorithms and present the results as a trivariate plot (or ternary) plot [30] in Fig. 6. In this plot, the locations of

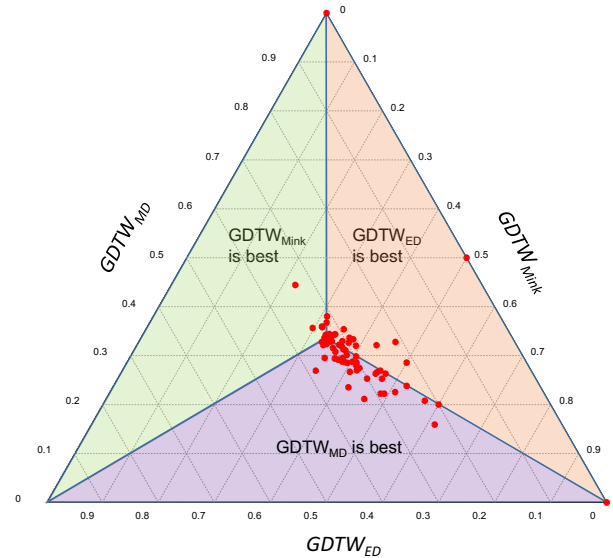


Fig. 6: A trivariate plot comparing $GDTW_{ED}$, $GDTW_{MD}$, and $GDTW_{Mink}$. For points close to the center of the figure, all 3 methods produce similar error rates. For points away from the center, at least one method performs poorly.

the points do not correspond to the actual error rate, but are *proportional* to them. This ternary plot shows the error rate distribution for the three distances and marks the areas where each individual distance performs better than the others. The red dots in the light purple area (bottom) indicate the wins for $GDTW_{MD}$ (in 56 cases), while the ones in the pink area (right) show where $GDTW_{ED}$ is better (in 44 cases) and the ones in the green area (left) correspond to $GDTW_{Mink}$ performing better (in 15 cases). The results are surprisingly diverse, with 44 “wins” for $GDTW_{ED}$, 56 “wins” for $GDTW_{MD}$, and 15 “wins” for $GDTW_{Mink}$. There are fewer dots in the green area, indicating that $GDTW_{Mink}$

generally performed poorer than the other two distances. The error rates for $GDTW_{MD}$ and $GDTW_{ED}$ are fairly close, as shown by the high concentration of points in the center.

In conclusion we succeeded at improving on the standard benchmark of 1-NN DTW, simply by switching to $GDTW_{MD}$. This suggests that using our framework to forge new time-warping distances has the potential to lead to other substantial improvements on state-of-the-art time series classification, upon which we elaborate in [25].

C. Best Match Retrieval and Clustering Experiments

For each dataset in a subset from the UCR archive, we randomly select a subsequence and “promote” it to be a query, similarly to [31]. Then we find the best match for this query sample by using three point-wise distances (ED , MD , $Mink$) and their warped counterparts ($GDTW_{ED}$, $GDTW_{MD}$, $GDTW_{Mink}$). We repeat this experiment for 10 random sample sequences. Due to the space constraints we show additional experimental results in [25], while here we offer only a summary analysis in Table I. The results vary

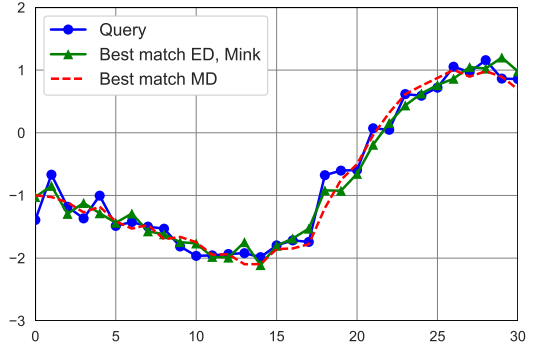
TABLE I: Percentage scenarios where pairs of GDTW variants return the same best match for a specific sample sequence in the ECG dataset

Pair of distances	Percent scenarios
ED and DTW	20
MD and $GDTW_{MD}$	20
$Mink$ and $GDTW_{Mink}$	0
DTW and $GDTW_{Mink}$	10
$GDTW_{MD}$ and $GDTW_{Mink}$	20
DTW and $GDTW_{MD}$	50

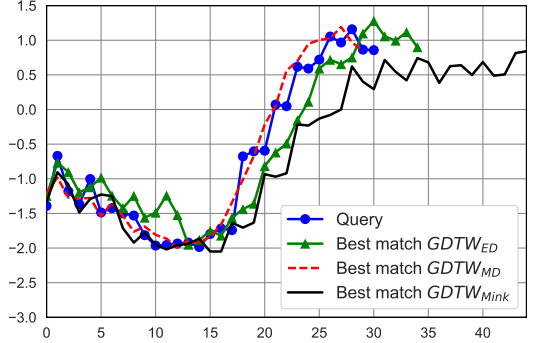
significantly when using different point-to-point distances and their warped versions, as expected. The point-wise distances can be at most as good as their warped versions, but only for sequences that are aligned in time. The $GDTW$ variants often return different results, each providing best matches that would otherwise be missed. Only in 10% of the scenarios did all three variants return the same best match.

In Fig. 7 we show a visual example of the best match in ECG retrieved by the point-wise and their respective warped counterpart distances. In this specific case, two of the point-wise distances returned the same best match (ED and $Mink$), while MD and each respective warped distance returned different matches. If our new distances would have all returned the same result or even the same result as their point-wise counterparts, then they would not be useful. It is the diversity of the results that proves the usefulness of these distances.

Lastly, we conduct an experiment that reveals new insights uncovered by our newly warped distances using **hierarchical clustering**. We select five sequences from the ECG dataset. Two of them are randomly chosen samples from class 1 (red/bold), while the other three are randomly selected from class -1 (blue/none-bold). We cluster these sequences using our three $GDTW$ variants. We repeat the experiment five times. As the dendrogram in Fig. 8 shows, $GDTW_{MD}$ clusters together sequences from the same class (red) from start,



(a) Best match for the same sample retrieved by ED, MD, Mink



(b) Best match for the same sample retrieved by DTW, $GDTW_{MD}$, $GDTW_{Mink}$

Fig. 7: Example of best match in ECG retrieved by point-wise distances (a) and their warped counterparts (b)

while $GDTW_{ED}$ and $GDTW_{Mink}$ do not, reaffirming that $GDTW_{MD}$ has a higher accuracy than DTW.

In short, our newly warped distances can reveal best matches missed by classic DTW and improve clustering quality.

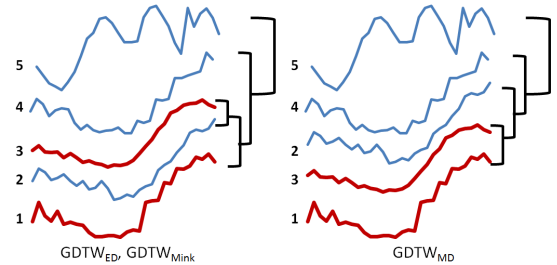


Fig. 8: Average linkage hierarchical clustering

D. Evaluating Warping Characteristics

Evaluating Cardinality of Warpings. We randomly select 20 pairs of sequences 10 pairs of the same and 10 pairs of different lengths. We find the matching elements of the sequences by using $GDTW_{ED}$, $GDTW_{MD}$, and $GDTW_{Mink}$.

We compute the average amount of warpings for each $GDTW$ variant for various datasets (See Table II). Generally, $GDTW_{Mink}$ and $GDTW_{MD}$ discover fewer warpings than the ones created by $GDTW_{ED}$. This indicates that these warped distances avoid singularities or “over-warping” incurred by classic DTW.

TABLE II: Average warpings for sequences of diverse lengths

Dataset	DTW	$GDTW_{MD}$	$GDTW_{Mink}$
ItalyPower	0.4	0.3	0.23
ECG	0.43	0.34	0.17
Wafer	0.49	0.38	0.33
Face	0.32	0.28	0.11

Summarizing our findings (full details and additional visual displays are available) [25], we offer a visual example of warping characteristics for a pair of sequences from the ItalyPower dataset in Fig. 9. Warpings indicate points of a sequence that are either matching or are being matched to more than one point of the other sequence. Points that are matched one-to-one are referred to as matchings. The warpings created by $GDTW_{MD}$ and $GDTW_{Mink}$ are fewer and more intuitive than the ones created by $GDTW_{ED}$, which is a similar conclusion with that of the experiments of [11]. This shows that indeed the classic DTW can “over-warp”, mainly due to the fact that it incorporates the ED as base distance. This knowledge can be useful to analysts who might choose to use distances that produce fewer warpings.

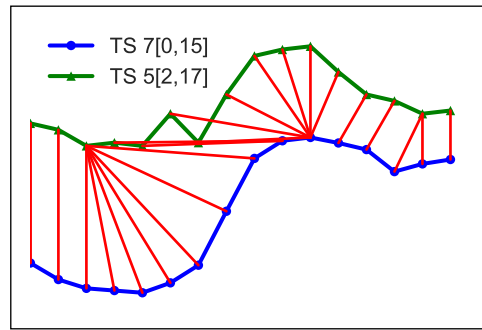
Evaluating Sensitivity to Local Distortions. We test the ability to find the correct warpings by using different $GDTW$ variants for pairs of sequences for which the “warping” is known. Similarly to [11], we distort the y-axis by adding or subtracting a distortion (Gaussian bump) on randomly chosen anchor points of the sequences. As shown in Fig. 11, each distance leads to a warping path different than the other two distances. DTW “over-warps” the distorted sequences, while $GDTW_{MD}$ and $GDTW_{Mink}$ find a shorter warping path. The performance for DTW and $GDTW_{MD}$ tends to degrade even for small distortions of the y-axis, while $GDTW_{Mink}$ maintains a better warping performance.

In summary, our experimental results confirm the utility of our newly warped distances (in particular, $GDTW_{Mink}$) in avoiding singularities or “over-warping” problems incurred by the classic DTW. They also are less sensible to distortion. In other words, they promise to be useful in practice.

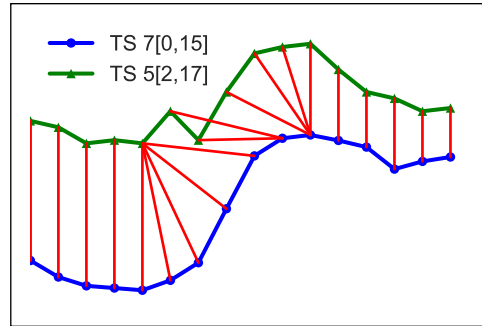
VII. STUDYING HEART ARRHYTHMIA USING GDTW

In collaboration with expert cardiologists, we explore the MIT-BIH Arrhythmia Database, created by Beth Israel Deaconess Medical Center and MIT, which supports research into arrhythmia analysis and related subjects. The MIT-BIH Arrhythmia Database [32], [33] contains 48 half-hour excerpts of two-channel ambulatory ECG recordings obtained from 47 subjects. 23 recordings were chosen at random from a set of 4000 24-hour ambulatory ECG recordings collected from a mixed population of inpatients (about 60%) and outpatients (about 40%) at Boston’s Beth Israel Hospital. To address the imbalance in the data, the remaining 25 recordings were selected from the same set to include less common but clinically significant arrhythmias that otherwise would not be well-represented in a small random sample.

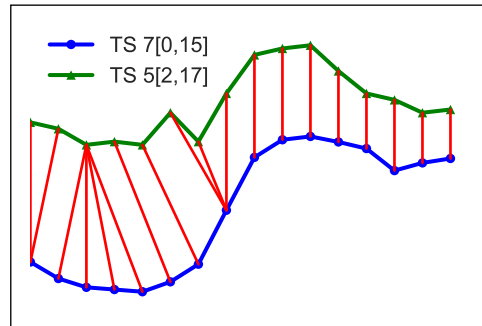
Medical staff studies similarity of ECGs for diagnosing arrhythmia which refers to changes of the normal sequence of electrical impulses. Electrical impulses may cause the heart



(a) Classic DTW warping



(b) $GDTW_{MD}$ warping



(c) $GDTW_{Mink}$ warping

Fig. 9: Visual warpings for a pair of sequences in ItalyPower

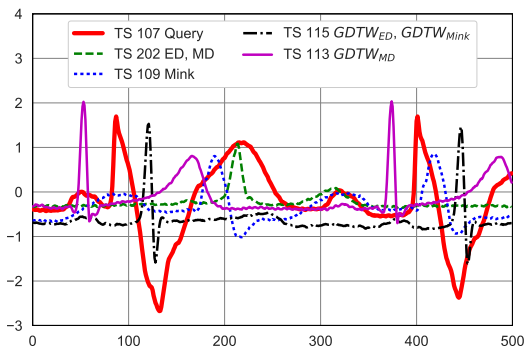
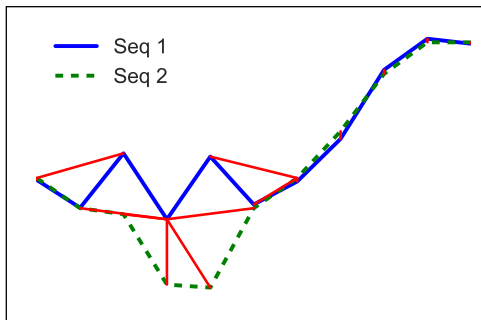
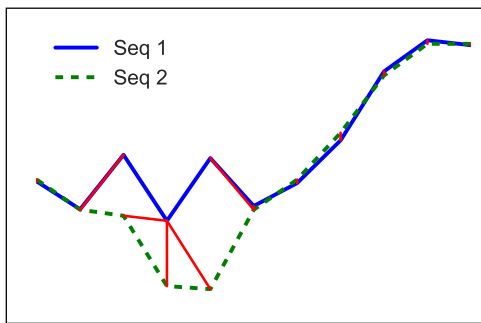


Fig. 10: Case study best match sequences for the same sample retrieved by GDTW variants

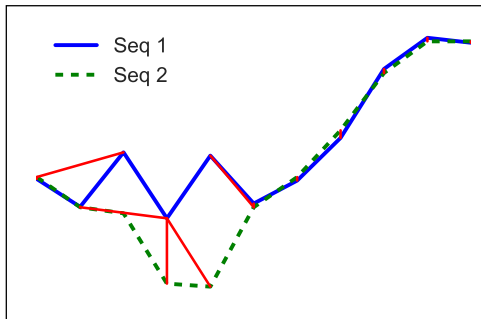
to beat too fast, too slowly, or erratically. When the heart does not pump blood effectively, the lungs, brain and other organs



(a) DTW warplings



(b) $GDTW_{MD}$ warplings



(c) $GDTW_{Mink}$ warplings

Fig. 11: **Warpings for distorted sequences.** (a), (b), and (c) show the warpings using respectively the classic DTW, $GDTW_{MD}$ and $GDTW_{Mink}$

cannot work properly and may shut down or become damaged.

We use our newly warped distances to explore this database and find the best match for a given ECG shape. For this experiment, we first chose the sample heart rate shape of the record labeled 107. This male patient (age 63) has a complete heart block condition in which the impulse generated in the sinoatrial node in the atrium of the heart does not propagate to the ventricles. We randomly selected 20 records from the dataset, including that of the patient with record 107 and asked our cardiologist collaborators to find the best match for this ECG shape. The cardiologists identified the ECG for the patient with record number 113, as having the closest heart rate, meaning average heart rate in beats per minute. Independent of their findings, we retrieved the best match for this sequence by using ED, MD, M_{ink} , $GDTW_{ED}$, $GDTW_{MD}$, and $GDTW_{Mink}$. As seen in Fig. 10, different distances

returned different best matches. Sequence 113 was returned as best match by $GDTW_{MD}$. All other distances returned different matches. In this case $GDTW_{MD}$ found the same best match as the domain expert. We repeated this experiment five times using different sample sequences and arrived to the same conclusion based on comparing the answers provided by the cardiologists with the ones retrieved by our system. Visual samples are archived along with our additional experimental results [25]. In short, in this ECG Arrhythmia use case, the newly warped $GDTW_{MD}$ consistently finds the best match confirmed by experts and missed by the classic DTW.

VIII. RELATED WORK

DTW has been popular in a large range of application domains including medicine [13], and spoken word recognition [20]. Despite its ability to compare mis-aligned time series, it can produce pathological results [11]. Many modifications of DTW have been proposed to improve the performance, produce better alignments, and to handle “singularities”. Most constrain the warpings, while continuing to use ED as base distance, as we describe below.

For performance improvement, Keogh and Pazzani [34] introduced PDTW, which applies the classic DTW algorithm to a higher level abstraction of the data (Piece Aggregate Approximation), outperforming DTW with little loss of accuracy. Indexing and optimization methods [7], [16], [17] further improved the response time in retrieving similar sequences.

Some works aim to address singularities and produce superior alignments by modifying the way the warping path is computed. Unfortunately they still keep ED as intrinsic base distance. For example, [35] introduced a variable penalty whenever a non-diagonal step is taken. This reduces the number of non-diagonal moves and improves the alignment of chromatogram signals. WDTW [19] penalizes points with a higher phase difference between a reference point and a testing point to prevent minimum distance distortion caused by outliers. Closer conceptually to our idea, [20] replaces ED with another base distance. However, it is restricted to only incorporating base distances based on sums, such as Manhattan. Our work is a major step forward, as our method is general enough to “warp” any distance, regardless of its mathematical expression.

Symmetric DTW [22] addresses slope weighting. In computer graphics, Iterative Motion Warping [18] finds a spatial temporal warping between two instances of motion captured data. In contrast, Derivative DTW [11] produces superior alignments by replacing ED with the square of the difference of the derivatives of the sequences, thus gaining more information about the shape. Closer to our framework, this replaces ED with a different base distance. Unlike our work, they stop at using only one derivative based distance, while our methodology incorporates a wide array of base distances. [36] integrates multiple “warped” distances such as LCSS [37], DTW and its variations like derivative DTW [11] for semi-supervised clustering. This framework combines multiple already defined “warped distances” while our framework pro-

vides strategies for warping point wise distances. [38] proposes a suite of operators for trajectories similarity based on locality, temporality, directionality, and rate of change. This work is specific to trajectory databases, whereas our framework aims to help mining time series database in general. [39] devises a new similarity measure capturing the delay of a reaction to an action between two time series. This extension of DTW [40] is compared to multiple matching methods (DTW, edit distance, LCFM [41]), but the combination of multiple metrics is not formalized nor connected to developing a methodology for defining a recurrence.

IX. CONCLUSION

Our proposed general time warping framework offers the first universal solution for transforming point-wise distances into robust alignment tools, capable of performing flexible sequence matching. Our methodology insures that warped distances can be designed and implemented in a consistent manner, establishing a valuable resource for the research community. This repository now includes warped versions of popular distances with complex mathematical expressions. While our paper demonstrates that distances warped by our GDTW methodology achieve improved accuracy over DTW for time series classification for 85 data benchmark data sets [10], [26], it opens the avenue for new research [25] in leveraging these *GDTW* variants for solving a broad range of problems from classification and clustering to singularities.

ACKNOWLEDGMENT

We thank NSF through grants IIS-1018443 and CRI-1305258 for financial support. Special thanks to Prof. Samuel Madden from MIT for invaluable feedback. We thank Dr. Masooma Shaukat from St. Bartholomew's Hospital, London for help with our arrhythmia case study. Thanks to M. Andrews and S. Swartz from WPI for help with the implementation.

REFERENCES

- [1] S.-H. Cha, "Comprehensive survey on distance/similarity measures between probability density functions," *City*, 2007.
- [2] E. Karakoc *et al.*, "Distance based algorithms for small biomolecule classification and structural similarity search," *Bioinformatics*, 2006.
- [3] J. Mason and D. Handscomb, *Chebyshev polynomials*. CRC Press, 2002.
- [4] M. Stricker and M. Orengo, "Similarity of color images," in *Symposium on Electronic Imaging: Science and Technology*. International Society for Optics and Photonics, 1995.
- [5] M. Swain and D. Ballard, "Color indexing," *International journal of computer vision*, 1991.
- [6] J. Smith, "Integrated spatial and feature image systems: Retrieval, analysis and compression," Ph.D. dissertation, Columbia Univ, 1997.
- [7] E. Keogh and C. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, 2005.
- [8] D. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series," in *KDD workshop*. Seattle, WA, 1994.
- [9] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*. ACM, 1994, vol. 23, no. 2.
- [10] L. Chen and R. Ng, "On the marriage of lp-norms and edit distance," in *Proceedings of the Thirtieth international conference on VLDB-Volume 30*. VLDB, 2004.
- [11] E. Keogh and M. Pazzani, "Derivative dynamic time warping." SIAM, 2001.
- [12] J. Aach and G. Church, "Aligning gene expression time series with time warping algorithms," *Bioinformatics*, 2001.
- [13] E. Caiani, A. Porta *et al.*, "Warped-average template technique to track on a cycle-by-cycle basis the cardiac filling phases on left ventricular volume," in *Computers in Cardiology 1998*. IEEE, 1998.
- [14] D. Gavrilu, "The visual analysis of human movement: A survey," *Computer vision and image understanding*, 1999.
- [15] B. Yi and C. Faloutsos, "Fast time sequence indexing for arbitrary lp norms." VLDB, 2000.
- [16] M. Vlachos, M. Hadjieleftheriou *et al.*, "Indexing multi-dimensional time-series with support for multiple distance measures," in *Proceedings of the ninth ACM SIGKDD*. ACM, 2003.
- [17] T. Rakthanmanon, Campana *et al.*, "Searching and mining trillions of time series subsequences under dynamic time warping," in *18th ACM SIGKDD*, 2012.
- [18] E. Hsu, K. Pulli, and J. Popovici, "Style translation for human motion," in *ACM Transactions on Graphics (TOG)*. ACM, 2005.
- [19] Y. Jeong, M. Jeong, and O. Omiaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognition*, 2011.
- [20] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 1978.
- [21] "Ucr time series classification archive," http://www.cs.ucr.edu/~eamonn/time_series_data/, 2015, [Online].
- [22] J. Kruskall and M. Liberman, "The symmetric time warping algorithm: From continuous to discrete. time warps, string edits and macromolecules," 1983.
- [23] R. Agrawal, C. Faloutsos, and A. Swami, *Efficient similarity search in sequence databases*. Springer, 1993.
- [24] J. Looman and J. Campbell, "Adaptation of sorensen's k (1948) for estimating unit affinities in prairie vegetation," *Ecology*, 1960.
- [25] "Additional gdtw material," <https://github.com/GDTW-Material/GDTWMaterial>, 2016, [Online].
- [26] Ding, Trajcevski *et al.*, "Querying and mining of time series data: experimental comparison of representations and distance measures," *Proceedings of the VLDB Endowment*, 2008.
- [27] A. Bagnall, A. Bostrom *et al.*, "The great time series classification bake off: An experimental evaluation of recently proposed algorithms. extended version," *arXiv preprint arXiv:1602.01711*, 2016.
- [28] X. Xi, E. Keogh *et al.*, "Fast time series classification using numerosity reduction," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 1033–1040.
- [29] M. Långkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.
- [30] "Wikipedia ternary plots," https://en.wikipedia.org/wiki/Ternary_plot.
- [31] A. Fu, E. Keogh *et al.*, "Scaling and time warping in time series querying," *The VLDB Journal*, 2008.
- [32] G. Moody and R. Mark, "The impact of the mit-bih arrhythmia database," *IEEE Engineering in Medicine and Biology Magazine*, vol. 20, no. 3, pp. 45–50, 2001.
- [33] A. Goldberger, L. Amaral, Glass *et al.*, "Physiobank, physiotoolkit, and physionet components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. e215–e220, 2000.
- [34] E. Keogh and M. Pazzani, "Scaling up dynamic time warping for datamining applications," in *Proceedings of the sixth ACM SIGKDD*. ACM, 2000.
- [35] D. Clifford, G. Stone *et al.*, "Alignment using variable penalty dynamic time warping," *Analytical chemistry*, 2009.
- [36] J. Zhou, S. Zhu *et al.*, "Enhancing time series clustering by incorporating multiple distance measures with semi-supervised learning," *Journal of Computer Science and Technology*, vol. 30, no. 4, pp. 859–873, 2015.
- [37] J. W. Hunt and T. G. Szymanski, "A fast algorithm for computing longest common subsequences," *Communications of the ACM*, vol. 20, no. 5, pp. 350–353, 1977.
- [38] N. Pelekis, I. Kopanakis, I. Ntoutsi, G. Marketos, and Y. Theodoridis, "Mining trajectory databases via a suite of distance operators," in *2007 IEEE 23rd International Conference on Data Engineering Workshop*. IEEE, 2007, pp. 575–584.
- [39] M. Konzack, T. McKetterick *et al.*, "Visual analytics of delays and interaction in movement data," *International Journal of Geographical Information Science*, vol. 31, no. 2, pp. 320–345, 2017.
- [40] J. Long and T. Nelson, "Measuring dynamic interaction in movement data," *Transactions in GIS*, vol. 17, no. 1, pp. 62–77, 2013.
- [41] K. Buchin, M. Buchin *et al.*, "Locally correct fréchet matchings," *Algorithms–ESA 2012*, pp. 229–240, 2012.