Fast Time Series Classification under Lucky Time Warping Distance

Stephan Spiegel, Brijnesh-Johannes Jain, Sahin Albayrak
DAI Lab, Berlin Institute of Technology
Ernst-Reuter-Platz 7, 10587 Berlin, Germany
{spiegel,jain,albayrak}@dai-lab.de

ABSTRACT

In time series mining, the Dynamic Time Warping (DTW) distance is a commonly and widely used similarity measure. Since the computational complexity of the DTW distance is quadratic, various kinds of warping constraints, lower bounds and abstractions have been developed to speed up time series mining under DTW distance.
In this contribution, we propose a novel Lucky Time Warping (LTW) distance, with linear time and space complexity, which uses a greedy algorithm to accelerate distance calculations for nearest neighbor classification. The results show that, compared to the Euclidean distance (ED) and (un)constrained DTW distance, our LTW distance trades classification accuracy against computational cost reasonably well, and therefore can be used as a fast alternative for nearest neighbor time series classification.

Categories and Subject Descriptors

G.3 [**Probability and Statistics**]: Time Series Analysis; I.5.3 [**Pattern Recognition**]: Clustering—*algorithms, similarity measures*

General Terms

Algorithms, Measurement, Performance

Keywords

Time Series, Distance Measures, Classification

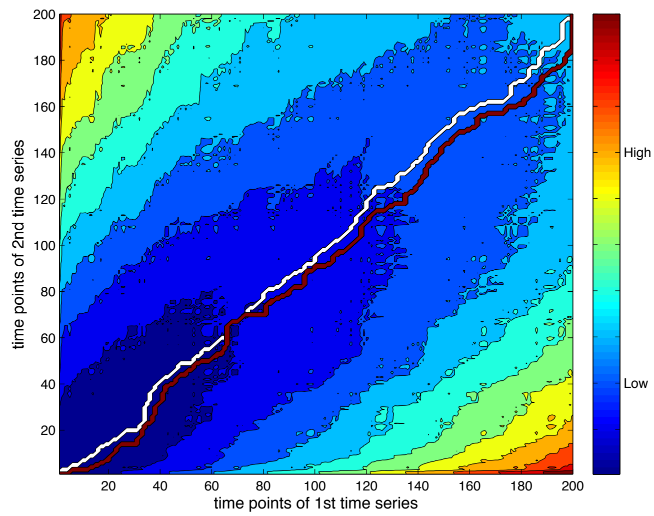

Figure 1: Warping matrix of two randomly generated time series, exemplifying the spread of the cumulative warping cost. The optimal (minimum cost) warping path computed by the brute-force dynamic programming approach of the DTW distance is shown in red color, and the lucky warping path found by the greedy approach of the LTW distance is shown in white color. Both paths pass through low cost regions (ranging from dark to light blue) and avoid high cost regions (displayed in shades of yellow and red).

1. INTRODUCTION

The Dynamic Time Warping (DTW) distance was first introduced to the data mining community almost two decades ago [2], and since then has been used as a utility for various time series mining tasks; including classification, clustering, and indexing of (sub)sequences [7]. Its current popularity and widespread use is owing to the fact that, in contrast to the Euclidean distance, the DTW distance works well for time series with various distortions and multiple invariance [1, 23]. But the brute-force dynamic programming approach of DTW leads to quadratic space and time complexity [3].
In time series classification, the combination of 1-Nearest-Neighbor (1NN) classifier with DTW has been proven exceptionally difficult to beat [24]. Nevertheless, if naively implemented, the 1NN-DTW approach is computationally demanding, because 1NN classification usually involves a great number of quadratic DTW calculations [19].
Hitherto existing approaches that aim to reduce the computational complexity of 1NN-DTW time series classification fall into the categories of algorithms [21, 22] which use (1) numerosity reduction to discard a large fraction of training data [5, 16, 24], (2) constraints to limit the number of cells that are evaluated in the cost/similarity matrix [8, 18, 19, 20], (3) abstraction to perform DTW on a reduced representation of the data [6, 10] or (4) lower bounding to reduce the number of times DTW must run [12, 14, 17, 25].

SAC'14, March 24-28, 2014, Gyeongju, Republic of Korea

http://dx.doi.org/10.1145/2554850.2554885.

We propose a novel Lucky Time Warping (LTW) distance, with linear time and space complexity, which uses a greedy algorithm to accelerate distance calculations for 1NN time series classification. The results show that, compared to constrained DTW with lower bounding, our LTW distance trades classification accuracy against computation time reasonably well, and therefore can be used as a fast alternative for nearest neighbor time series classification. Figure 1 allows a geometric intuition for the optimal/minimum cost and lucky warping path returned by the DTW and LTW distance function respectively.

Our work was inspired by other alternative procedures for implementing a DTW algorithm with substantially reduced computation, including early techniques for a directed graph search through a grid to find the best warping path [4] and recently proposed greedy approximate solutions [15]. But, to the best of our knowledge, there is no study that thoroughly compares the performance of the traditional DTW distance (with/out constraint and lower bound) to an alternative greedy approximate solution, like our proposed LTW distance, in respect of classification accuracy and computation complexity, on an extensive time series archive [13], as presented in this work.

The rest of the paper is organized as followed: Section 2 introduces notation and background to allow a formal definition of our LTW distance in Section 3. Experimental results on classification accuracy and computational cost of our proposed LTW distance are presented in Section 4. We conclude with a discussion of future work in Section 5.

2. NOTATION AND BACKGROUND

The 1NN-DTW approach for time series classification has been shown to achieve high classification accuracy, and its computational demand has been mitigated by warping constraints and pruning techniques. Hence, we want to give some background on DTW and related speedup techniques, before we introduce and compare our own LTW distance.

2.1 Dynamic Time Warping

The DTW algorithm aims to find the optimal warping path with minimum cost to describe the similarity of two time series. In case that the optimal warping path amounts to a relatively low cost the examined time series are considered to be similar, whereas high cost or distance implies dissimilarity [2]. To determine the (dis)similarity of two time series or rather to find the optimal warping path the DTW algorithm uses a brute-force dynamic programming approach which tests all possible paths within the defined warping window, leading to quadratic complexity [3].

Since many of the subsequent explanations are based on DTW, we present a formal definition of the DTW distance as introduced by Keogh et al. [10, 11]. Suppose we have two time series Q and C, of length n and m respectively, where:

$$\begin{aligned} Q &= (q_1, q_2, \ldots, q_i, \ldots, q_n) \\ C &= (c_1, c_2, \ldots, c_j, \ldots, c_m) \end{aligned} \quad (1)$$

To align these two sequences, the DTW algorithm first constructs a n-by-m matrix, where the (i^{th} , j^{th}) element of the matrix corresponds to the squared distance, $d(q_i, c_j) = (q_i - c_j)^2$, which is the alignment between points q_i and c_j. To find the best match between these two sequences, the DTW algorithm retrieves a path through the matrix that minimizes the total cumulative distance between them. In particular, the optimal path is the path that minimizes the warping cost:

$$DTW(Q, C) = \min \left\{ \sqrt{\sum_{k=1}^{K} w_k} \right\} \quad (2)$$

where w_k is the matrix element $(i, j)_k$ that also belongs to k^{th} element of a warping path W, a contiguous set of matrix elements that represent a mapping between Q and C. This warping path can be found using dynamic programming to evaluate the following recurrence.

$$\gamma(i, j) = d(q_i, c_j) + \min\{\gamma(i\text{-}1, j\text{-}1), \gamma(i\text{-}1, j), \gamma(i, j\text{-}1)\} \quad (3)$$

where $d(i, j)$ is the distance found in the current cell, and $\gamma(i, j)$ is the cumulative distance of $d(i, j)$ and the minimum cumulative distances from the three adjacent cells. Moreover, the warping path is typically subject to several constraints.

- The **Boundary conditions** require the warping path start and finish in diagonally opposite corner cells of the matrix, with $w_1 = (1, 1)$ and $w_K = (m, n)$.
- The **Continuity conditions** restrict the allowed steps in the warping path to adjacent cells, including diagonally adjacent cells. Given $w_k = (a, b)$ then $w_{k-1} = (a', b')$ where $a - a' \leq 1$ and $b - b' \leq 1$.
- The **Monotonicity conditions** force the points in W to be monotonically spaced in time. Given $w_k = (a, b)$ then $w_{k-1} = (a', b')$ where $a - a' \geq 0$ and $b - b' \geq 0$.

2.2 Numerosity Reduction

One way to reduce the classification time of the 1NN-DTW approach is to discard a large fraction of the training data by means of numerosity reduction algorithms [5, 24], which choose the objects that maintain or even improve the accuracy. Other work on dissimilarity-based classifiers even suggests that by using a few, but well chosen prototypes, it is possible to achieve a better classification performance in both speed and accuracy than by using all the training samples [16].

2.3 Warping Constraints

Constraints work well in domains where time series have only a small variance, but perform poorly if time series are of events that start and stop at radically different times [22]. The two most commonly used constraints are the Sakoe-Chiba Band [20] and the Itakura Parallelogram [8], which both speed up DTW by a constant factor, but still lead to quadratic complexity if the window size is a linear function of the time series length [19, 22]. Given the formal definition of DTW in Section 2.1, the Sakoe-Chiba Band [20] is an:

> **Adjustment Window condition** which corresponds to the fact that time-axis fluctuation in usual cases never causes a too excessive timing difference. Given $w_k = (a, b)$ then $|a - b| \leq r$ where r is the size of the warping window.

Recent work has shown that learned constraints are able to boost classification accuracy and outperform any uniform band [18]. Although optimal band shape allows tighter lower bounds and thus more effective pruning, the Euclidean distance is still orders of magnitude faster than DTW with learned constraints [18].

2.4 Abstraction

Early work on time series representation introduces a modification of DTW which operates on a higher level abstraction, and produces one to two orders of magnitude speedup with no decrease in accuracy [6, 10].
Although abstraction speeds up DTW by operating on a reduced presentation of the data, the calculated warp path becomes increasingly inaccurate as the level of abstraction increases [22], and projecting the low resolution warp path to the full resolution usually gives a result far from optimal, because significant local variations are ignored.

2.5 Lower Bounding

Another way to mitigate the high computational complexity of 1NN-DTW time series classification is to prune similarity calculations by lower bounding the DTW distance measure. Recent studies on large-scale time series data have shown that DTW in combination with certain lower bounds lead to quasi-linear complexity for indexing problems [17, 19], leading to the widespread belief that there is no need for further improvements in speed. However, pruning does not effect the intrinsic quadratic complexity of the actual DTW calculation [22].
There are two desirable properties of a lower bounding measure; it must (1) be fast to compute and (2) return a relatively close approximation of the true distance [12, 18, 19]. Since the aim of lower bounding is to prune computational expensive similarity calculations, an approximation is required to be faster than the original measure and should be at most linear in the length of the sequences.
Known approaches to lower-bound the DTW distance include approximations that involve the difference between minimum and maximum points [14, 25] and more sophisticated techniques that employ global constraints to create warping-envelopes [12].
For further explanations and experiments we consider the lower bound introduced by Keogh and Ratanamahatana [12], denoted as LB_Keogh, which compares each candidate time series C to the upper U and lower L sequence of the bounding envelope that encloses the time series query Q. In case of the Sakoe-Chiba band with an window size r the two new sequences can be defined as:

$$\begin{aligned} U_i &= \max(q_{i-r} : q_{i+r}) \\ L_i &= \min(q_{i-r} : q_{i+r}) \end{aligned} \tag{4}$$

where $\forall\, U_i \geq q_i \geq L_i$. Having defined U and L, the lower bounding measure for DTW can be formulated as:

$$LB\_Keogh(Q,C) = \sqrt{\sum_{i=1}^{n} \begin{cases} (c_i\text{-}U_i)^2 & \text{if } c_i{>}U_i \\ (c_i\text{-}L_i)^2 & \text{if } c_i{<}L_i \\ 0 & \text{otherwise} \end{cases}} \tag{5}$$

For any two sequences Q and C of the same length n, for any global constraint on the warping path of the form $j - r \leq i \leq j + r$, the following inequality holds:

$$LB\_Keogh(Q,C) \leq DTW(Q,C) \tag{6}$$

2.6 Amortized Computational Cost

Since execution time is a poor choice for evaluating the performance gain achieved by lower bounds, recent work [11, 12] suggest to compute the average or amortized percentage of warping matrix that need to be accessed for each individual DTW distance calculation.
In order to compare the computational efficiency of 1NN time series classification in relation to arbitrary distance measures and pruning techniques, we propose to compute the amortized computational cost (ACC) by accounting the basic similarity operations of the classification task with regard to the time series length. We define a basic similarity operation as the squared distance between a pair of time points, which corresponds to the calculation of exactly one entry in the time warping matrix:

$$d(q_i, c_j) = (q_i - c_j)^2 \tag{7}$$

Assuming two time series of length n and a warping constraint of r time points (Sakoe-Chiba band), we can compute the number of similarity operations or rather matrix elements E that are involved in a single cDTW (constrained Dynamic Time Warping) distance calculation by the following equation:

$$E_{cDTW}(n,r) = (2r+1)n - r(r+1) \tag{8}$$

In case that cDTW is combined with lower bounding (LB) techniques, we can reduce the number of expensive cDTW similarity calculations at the ratio of all 1NN comparisons to p percent. Due to the fact that a lower bound approximation is computed for every similarity comparison, no matter if an expensive similarity calculation follows or not, one may assume an additional computational cost that is constant. The computational cost for LB_Keogh [12] equates to:

$$E_{\text{LB-Keogh}}(n) = 2n \tag{9}$$

since each candidate time series is compared to the upper and lower sequence of the bounding envelope that encloses the time series query. The comparison between candidate time series and bounding envelope is based on the Euclidean distance which involves $2n$ squared distance computations or basic similarity operations. Consequently, one can compute the amortized computational cost of cDTW with LB_Keogh as follows:

$$\begin{aligned} E_{\text{cDTW-LB}}(n,r) &= E_{\text{LB-Keogh}}(n) + p * E_{cDTW}(n,r) \\ &= 2n + p * \Big((2r+1)n - r(r+1)\Big) \end{aligned} \tag{10}$$

In Section 4.4 we evaluate the ACC of 1NN classification in combination with DTW and LB_Keogh (1NN-DTW-LB approach) for a variety of time series datasets.

3. LUCKY TIME WARPING DISTANCE

In this section we introduce a novel Lucky Time Warping (LTW) distance, explain its underlying algorithm to find a warping path, and analyze its theoretical minimum and maximum computational cost.
The LTW distance finds a warping path that is constrained by the same (Boundary, Continuity, and Monotonicity) conditions that apply to the DTW distance. But, unlike the DTW distance which computes all elements within the warping matrix or rather warping window to find the minimum cost warping path, the LTW distance uses a greedy approach that only evaluates matrix elements which most likely contribute to the actual warping path. According to the formalization in Section 2.1, the LTW distance between two time series Q and C is defined as:

$$LTW(Q,C) = \sqrt{\sum\nolimits_{k=1}^{K} w_k} \tag{11}$$

Algorithm 1 LTW Distance Measure

Input: $Q, C \ldots$ time series; $w \ldots$ warping window
Output: $d \ldots$ lucky distance

1: $i, j \leftarrow 1$
2: $d \leftarrow (q_i - c_j)^2$ $\{q_i, c_j$ equals $Q(i), C(j)\}$
3: $n \leftarrow$ length of Q
4: $m \leftarrow$ length of C
5: **while** $(i \leq n)$ and $(j \leq m)$ **do**
6:   **if** $(i+1 \leq n)$ and $(j+1 \leq m)$ **then**
7:     $d_{dia} \leftarrow (q_{i+1} - c_{j+1})^2$
8:   **end if**
9:   **if** $(i+1 \leq n)$ and $(|i+1-j| \leq w)$ **then**
10:     $d_{up} \leftarrow (q_{i+1} - c_j)^2$
11:   **end if**
12:   **if** $(j+1 \leq m)$ and $(|j+1-i| \leq w)$ **then**
13:     $d_{right} \leftarrow (q_i - c_{j+1})^2$
14:   **end if**
15:   $d_{\min} = \min(d_{dia}, d_{up}, d_{right})$
16:   $d \leftarrow d + d_{\min}$
17:   $i, j \leftarrow \text{index}(d_{\min})$ {update position}
18: **end while**

where w_k is the k^{th} element of a warping path W. Given w_k with $k = (i, j)$ then the warping path can be found using the following recurrence.

$$w_{k+1} = \min\{d(q_{i+1}, c_{j+1}), d(q_{i+1}, c_j), d(q_i, c_{j+1})\} \quad (12)$$

where ties are resolved in the order as given in the list of arguments. Imposing an order on picking the next warping w_{k+1} ensures that the LTW distance is well-defined. Since

$$LTW(Q, C) = LTW(C, Q) \quad (13)$$

only holds if there are no ties, i.e. $d_{dia} \neq d_{up} \neq d_{right}$ (see Algorithm 1), we find that the LTW distance is not symmetric, and consequently not a metric. From

$$LTW(Q, C), LTW(C, Q) \geq DTW(Q, C) \quad (14)$$

follows that the LTW distance is an upper bound of the DTW distance. Since global constraints such as the Sakoe-Chiba band and the Itakura parallelogram can be imposed on the LTW distance in the usual way, the upper bounding property holds true in the constrained case. We deliberately refrain of discussing the tightness of LTW in respect to DTW, as will be explained in Section 4.5.

3.1 Lucky Warping Path

In contrast to DTW's brute-force approach, our LTW distance determines the warping path in a greedy manner, performing a variation of best-first search within the defined warping window. Consequently, the warping path computed by LTW is contained in the set of paths examined by DTW, just like the Euclidean distance is a special case of DTW where no warping takes place. Since DTW always finds the optimal warping path with minimum cost, our proposed LTW distance is supposed to return a suboptimal path with higher or equal cost/distance.

Figure 1 illustrates the optimal and lucky warping path corresponding to the DTW and LTW distance of two randomly generated time series. Furthermore, Figure 1 exemplifies the spread of the cumulative warping cost, and shows that both optimal and lucky warping path only pass through low cost regions and avoid high cost regions. The observation of Figure 1, and other experiments on random data which we omit

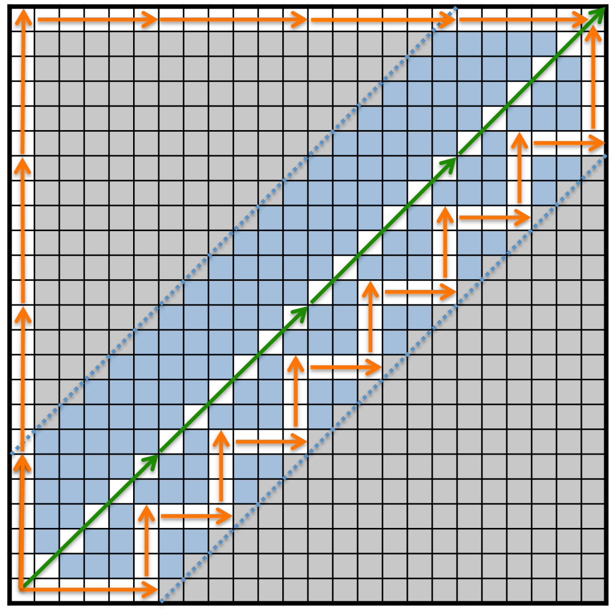

Figure 2: The shortest and longest possible warping path, constructed by the greedy algorithm of our LTW distance, highlighted in green and orange color respectively. The warping window is shown in blue.

for sake of brevity, give reason to expect the lucky warping path to be close to the optimal warping path, or even to overlap with the minimum cost path in some time intervals. Nevertheless, as will be explained in Section 4.5, we refrain of discussing the tightness of LTW in respect to DTW.

3.2 Greedy Algorithm

In contrast to the brute-force dynamic programming approach of DTW, our LTW distance determines the warping path in a greedy manner, performing a kind of best-first search through the similarity matrix.

As described in Algorithm 1, the LTW approach constructs a warping path by iteratively moving from one corner to the opposite corner of the similarity matrix (*Line 5*), repeatedly evaluating the cost of adjacent cells (*Line 7,10, and 13*) - going one step diagonal, up and right (*Line 6, 9, and 12*). Since a warping path must be monotonically spaced in time, the direction of the iterative moves through the similarity matrix is restricted, preventing backward movement. In fact, our LTW distance meets the Boundary, Continuity, and Monotonicity condition as formalized in Section 2.1.

3.3 Computational Cost

In the following section, we examine the computational cost of our proposed LTW distance measure from a theoretical perspective.

The introduced greedy algorithm constructs a warping path in an iterative manner, with each iteration involving the cost evaluation of three adjacent cells in the similarity matrix, one for each direction of the next possible move. Algorithm 1 demonstrates that the evaluation of an individual cell is associated with one call of the squared distance function (*Line 7, 10, and 13*), which represents our basic similarity operation (see Equation 7).

Consequently, the theoretical computational cost of one LTW distance calculation is three times the length of the corre-

sponding warping path. But it is important to understand that with zero warping the cost degrades to one times the length of the warping path, because our LTW distance only needs to evaluate the cells on the main diagonal, essentially returning the Euclidean distance.
Figure 2 gives an intuition for the minimum and maximum computational cost with and without warping constraint. Assuming that we want to measure the distance between two time series of length n, and knowing that the shortest possible warping path corresponds to the main diagonal of the similarity matrix, we expect LTW's greedy algorithm to take at least n iterative steps to compute the lucky warping path, leading to a minimum computational cost of $E_{LTW}(n) = 3n$ with warping window and $E_{LTW}(n) = n$ with zero warping. Furthermore, Figure 2 illustrates that the longest possible warping path (that starts and finishes in diagonally opposite corner cells of the similarity matrix) equals the sum of the time series lengths, meaning the LTW algorithm exhibits a maximum computation cost of $E_{LTW}(n) = 3(n + n) = 6n$, which is essentially linear. Nevertheless, the worst case is likely to happen with and without warping window, since $E_{LTW}(n) = E_{cLTW}(n)$ as illustrated in Figure 2.

4. EMPIRICAL EVALUATION

The goal of our comparative evaluation is twofold: (i) we investigate how well the LTW distance is suited for 1NN classification; and (ii) we assess the average amortized computational cost of the greedy algorithm for computing the LTW distance.

4.1 Time Series Data

For our empirical evaluation we consider 43 datasets of time series with different length and number of classes, as well as different properties of noise, auto-correlation and stationarity [13]. Each dataset is split into a training and test set, which are used for distance/parameter learning and evaluation respectively.

4.2 Experimental Protocol

For the sake of comparison, all our experiments on classification performance involved the 1NN classifier. We compared the classification accuracy of the 1NN approach applying the following distance measures:

- cLTW - Lucky Time Warping distance with warping constraint (Sakoe-Chiba band)
- ED - Euclidean distance - baseline measure [9]
- DTW - Dynamic Time Warping distance without warping constraint (100% warping window)
- cDTW - Dynamic Time Warping distance with warping constraint (Sakoe-Chiba band) - state-of-the-art approach [17]

To compute the distances, we applied the greedy algorithm for cLTW and dynamic programming for DTW and cDTW. We learned the bandwidth of the Sakoe-Chiba band for cLTW and cDTW by leave-one-out cross validation on the training set of the respective dataset. We chose the warping window size that gives the highest classification accuracy (see detailed results in Figure 5).
Moreover, we compared the ACC of cLTW and cDTW_LB - constrained Dynamic Time Warping with lower bounding, in particular LB_Keogh [12] as formulated Section in 2.5. We considered ACCs instead of clock times to avoid implementation bias, caused by the disparity of algorithmic design of the competing approaches.
Please note that we deliberately refrain from investigating early abandoning [17], because this speed-up technique can be applied for both DTW and LTW distance in the same way. In case of LTW, during the computation of the lucky warping path, if we note that the current sum of the squared differences between each pair of corresponding data points exceeds the best-so-far nearest neighbor, then we can stop the calculation [17]. Be that as it may, early abandoning has no influence on the classification accuracy.

4.3 Classification Accuracy

Figure 3 illustrates the classification accuracy of cLTW in comparison to ED, DTW and cDTW. Each scatter plot in Figure 3 summarizes the classification results of two contrasted measures, where each data point demonstrates the achieved accuracy for one individual time series dataset. Data points that lie on the upper side of the main diagonal (highlighted in red) illustrate a superior classification accuracy of the LTW algorithm, whereas data points on the lower side of the main diagonal indicate better performance of the compared distance measure. In case that a data point is plotted on or very close to the main diagonal, we can infer that both measures achieved the same or similar results on the corresponding dataset.
As we can see in the first scatter plot of Figure 3, the majority of data points (37/43) lie on the upper side of the main diagonal, meaning cLTW outperforms ED in terms of classification accuracy. From the second plot in Figure 3 we can infer that cLTW performs slightly better than DTW without warping constraint, because almost all data points are arranged close to the main diagonal, some lie on the diagonal, and half of them (21/43) lie on the upper side. That cLTW is superior to DTW can be confirmed by the summary given in Table 1, which shows that cLTW has a lower average classification error. From the third plot in Figure 3 we can observe that more data points indicate superior performance of cDTW, although cLTW achieves better classification results on a considerable number (16/43 excluding ties) of the examined datasets.
The detailed classification results of the considered distance measures on all examined datasets can be found in Figure 5. A summary is presented in the following Table 1.

	min	max	avg	std
ED	0.0050	0.6580	0.2541	0.1510
DTW	0.0000	0.6230	0.2113	0.1550
cDTW	0.0015	0.6130	0.1911	0.1453
cLTW	0.0020	0.6299	0.2033	0.1550

Table 1: Minimum, maximum, average, and standard deviation of classification errors for all four considered measures.

4.4 Amortized Computation Cost

Figure 4 shows the amortized computational cost (ACC) ratio of cDTW_LB to cLTW, where each data point compares the efficiency on one individual time series dataset. An

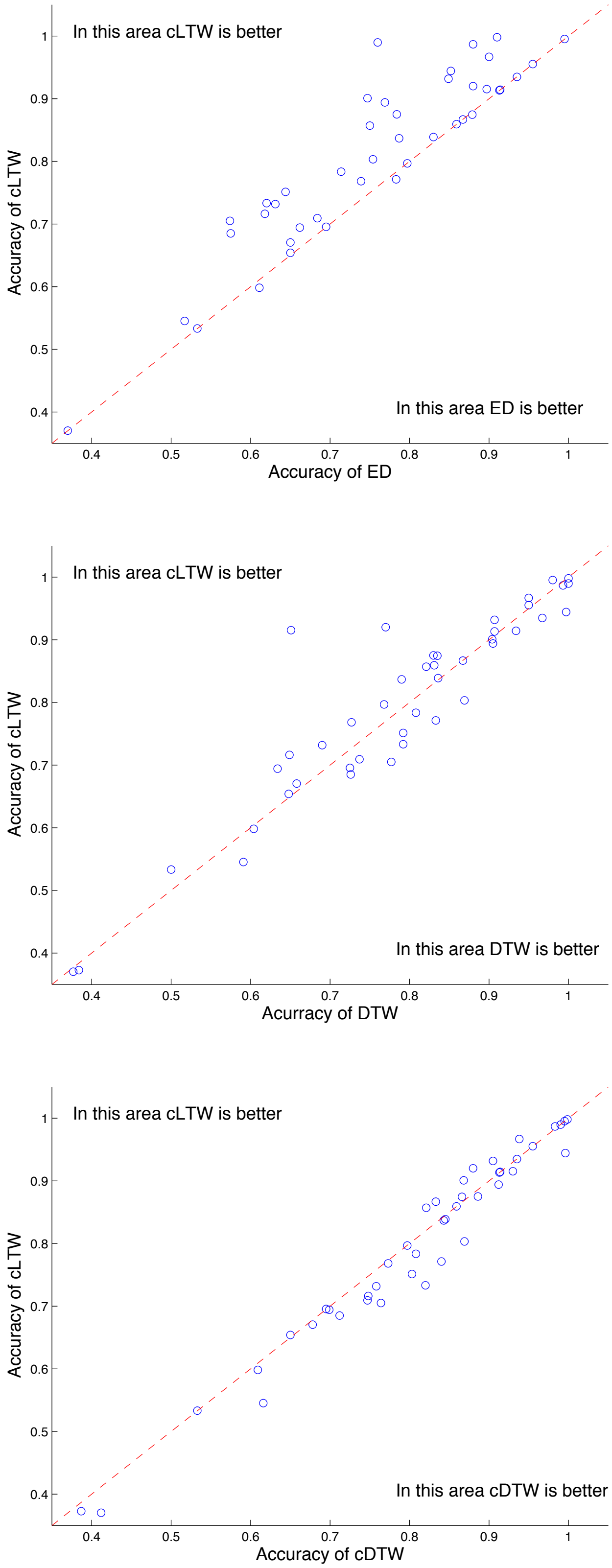

Figure 3: Classification accuracy of our introduced cLTW distance measure on all 43 considered time series datasets, listed in Figure 5, compared to ED, DTW and cDTW. In summary, cLTW achieved higher accuracy in 37/43, 21/43, and 16/43 cases, excluding ties that lie on the diagonal.

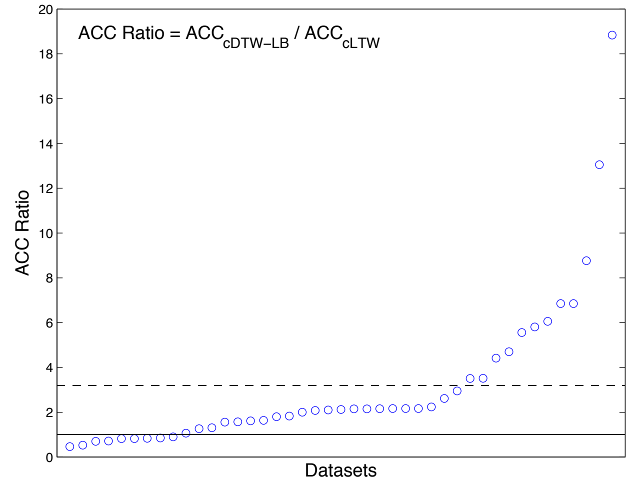

Figure 4: ACC ratio of cDTW_LB to LTW. Each data point demonstrates the contrasted efficiency on one time series dataset. The vertical axis shows how much LTW is faster than cDTW_LB. Results are arranged in ascending order along the horizontal axis. The black solid line indicates an ACC ratio of 1 or rather same efficiency. The average ACC ratio or speed-up of 3.2 is shown by the black dashed line.

ACC ratio of 1, indicated by the black solid line in Figure 4, means that both distance measures achieved the same efficiency or average amortized computational cost. Data points that lie above this baseline denote a higher efficiency or rather lower amortized computational cost of our cLTW distance. Figure 4 reveals that in some cases cLTW achieves an efficiency gain by one order of magnitude. On average cLTW is about 3.2 times faster than cDTW_LB, indicated by the black dashed line in Figure 4.
Detailed results on the average ACC of cLTW and cDTW_LB on all examined time series datasets are listed in Figure 5. A summary of the ACC results is presented in Table 2.

	min	max	avg	std
cDTW_LB	$2.02n$	$92.75n$	$10.89n$	$15.69n$
cLTW	$0.96n$	$4.94n$	$3.29n$	$1.39n$

Table 2: Minimum, maximum, average, and standard deviation of ACC results for cDTW_LB and cLTW in $E(n) = k \times n$ notation, where n is the time series length and k is the respective constant factor.

The ACC results for cLTW are within the range of $n \leq E_{cLTW}(n) \leq 5n$, which corresponds to our theoretical minimum and maximum computational cost derived in Section 3.3. In contrast, the ACC results of cDTW_LB are within the range of $2n \leq E_{\text{cDTW-LB}}(n) \leq 93n$, depending on the size of the warping window and the amount of pruned DTW calculations (see Equation 10).
Based on the presented ACC results we can infer that the efficiency of cDTW_LB varies strongly with the data under study, whereas cLTW is more robust in terms of required similarity operations. Due to the fact that the computational cost of cLTW does not depend on warping constraints or pruning power, it is an excellent choice for 1NN classification of arbitrary time series.

UCR DATASET	Classification Error in Percent of Misclassified Time Series 1NN-ED	1NN-DTW	1NN-cDTW (x%)	1NN-cLTW (x%)	ACC in E(n)=k*n Notation 1NN-cDTW-LB	1NN-cLTW
50words	0.3690	0.3100	**0.2420** (6)	0.2681 (3)	4.85 n	**3.82** n
Adiac	**0.3890**	0.3960	0.3910 (3)	0.4015 (3)	7.27 n	**3.46** n
Beef	**0.4670**	0.5000	**0.4670** (0)	**0.4667** (0)	2.15 n	**1.00** n
CBF	0.1480	**0.0030**	0.0040 (11)	0.0556 (3)	19.98 n	**4.53** n
ChlorineConcentration	0.3500	0.3520	0.3500 (0)	**0.3458** (2)	**2.02** n	4.28 n
CinC ECG torso	0.1030	0.3490	**0.0700** (1)	0.0848 (2)	10.12 n	**4.71** n
Coffee	0.2500	0.1790	0.1790 (3)	**0.1429** (2)	7.22 n	**4.01** n
Cricket X	0.4260	**0.2230**	0.2360 (7)	0.2949 (3)	7.60 n	**4.87** n
Cricket Y	0.3560	0.2080	**0.1970** (17)	0.2487 (2)	26.19 n	**4.71** n
Cricket Z	0.3800	0.2080	**0.1800** (7)	0.2667 (4)	7.78 n	**4.94** n
DiatomSizeReduction	0.0650	**0.0330**	0.0650 (0)	0.0654 (0)	2.23 n	**1.00** n
ECG200	0.1200	0.2300	0.1200 (0)	**0.0800** (2)	**2.05** n	3.84 n
ECGFiveDays	**0.2030**	0.2320	**0.2030** (0)	**0.2033** (0)	2.15 n	**0.99** n
FaceAll	0.2860	**0.1920**	**0.1920** (3)	0.2166 (1)	5.88 n	**3.65** n
FaceFour	0.2160	0.1700	**0.1140** (2)	0.1250 (1)	11.05 n	**3.74** n
FacesUCR	0.2310	0.0951	**0.0880** (12)	0.1059 (3)	24.45 n	**4.04** n
FISH	0.2170	0.1670	**0.1600** (4)	0.2286 (2)	12.00 n	**3.41** n
Gun Point	**0.0870**	0.0930	**0.0870** (0)	**0.0867** (0)	2.11 n	**0.99** n
Haptics	0.6300	0.6230	**0.5880** (2)	0.6299 (0)	5.80 n	**1.00** n
InlineSkate	0.6580	0.6160	**0.6130** (14)	0.6273 (1)	92.75 n	**4.92** n
ItalyPowerDemand	**0.0450**	0.0500	**0.0450** (0)	**0.0447** (0)	2.07 n	**0.96** n
Lighting2	0.2460	**0.1310**	**0.1310** (6)	0.1967 (4)	31.39 n	**4.58** n
Lighting7	0.4250	**0.2740**	0.2880 (5)	0.3151 (4)	11.75 n	**4.49** n
MALLAT	0.0860	**0.0660**	0.0860 (0)	0.0857 (0)	2.08 n	**1.00** n
MedicalImages	0.3160	0.2630	**0.2530** 20	0.2908 (2)	16.05 n	**3.41** n
MoteStrain	**0.1210**	0.1650	0.1340 (1)	0.1254 (1)	3.19 n	**2.99** n
OliveOil	**0.1330**	**0.1330**	0.1670 (1)	**0.1333** (0)	13.03 n	**1.00** n
OSULeaf	0.4830	0.4090	**0.3840** (7)	0.4545 (3)	14.54 n	**4.14** n
SonyAIBORobotSurface	0.3050	**0.2750**	0.3050 (0)	0.3045 (0)	2.14 n	**0.99** n
SonyAIBORobotSurfaceII	**0.1410**	0.1690	**0.1410** (0)	**0.1406** (0)	2.13 n	**0.98** n
StarLightCurves	0.1510	0.0930	0.0950 (16)	**0.0681** (23)	41.42 n	**4.73** n
SwedishLeaf	0.2130	0.2100	**0.1570** (2)	0.1632 (1)	**2.81** n	3.28 n
Symbols	0.1000	0.0500	0.0620 (8)	**0.0332** (9)	26.35 n	**3.85** n
synthetic control	0.1200	**0.0070**	0.0170 (6)	0.0133 (11)	7.26 n	**4.43** n
Trace	0.2400	**0.0000**	0.0100 (3)	0.0100 (3)	**3.98** n	4.82 n
Two Patterns	0.0900	**0.0000**	0.0015 (4)	0.0020 (7)	**2.93** n	4.11 n
TwoLeadECG	0.2530	**0.0960**	0.1320 (5)	0.0992 (7)	7.37 n	**4.02** n
uWaveGestureLibrary X	0.2610	0.2730	**0.2270** (4)	0.2317 (3)	**3.07** n	3.40 n
uWaveGestureLibrary Y	0.3380	0.3660	**0.3010** (4)	0.3057 (3)	**2.80** n	3.39 n
uWaveGestureLibrary Z	0.3500	0.3420	**0.3220** (6)	0.3294 (1)	3.85 n	**2.93** n
wafer	0.0050	0.0200	0.0050 (1)	**0.0047** (1)	**2.08** n	2.97 n
WordsSynonyms	0.3820	0.3510	**0.2520** (8)	0.2837 (4)	7.60 n	**3.81** n
Yoga	0.1700	0.1640	**0.1550** (2)	0.1613 (1)	**2.68** n	3.20 n

Figure 5: Detailed experimental results on classification error and amortized computational cost (ACC) for our proposed LTW distance on all examined datasets. For the sake of completeness, we list the classification errors of ED, DTW and cDTW published on the UCR Time Series Classification/Clustering Page [13]. Still, all prior results were verified and reproduced by our own implementation. For each dataset the lowest achieved classification error and amortized computational cost are printed in bold font. Rows with gray background indicate the datasets were LTW achieved both lower classification error and lower average ACC. Please note that the respective best/learned warping window size for each dataset is listed in brackets behind the classification accuracies. The ACC results assume the same window size for each approach respectively.

4.5 Discussion

Our empirical evaluation reveals that in some cases LTW achieved higher classification accuracy than DTW and/or cDTW, meaning that an optimal warping path strategy may lead to suboptimal classification results. In other words, an optimal warping path, as returned by DTW, does not necessarily lead to a low classification error, and an approximate suboptimal warping path, as returned by LTW, may contribute to an overall higher classification accuracy. Intuitively this statement can be justified by the fact that an optimal warping path does not necessarily minimize the classification error. Practically this statement can be confirmed by the finding [13] that ED is able to achieve higher classification accuracy than DTW and/or cDTW (see results for Adiac and MoteStrain dataset in Figure 5). Due to this statement, there is no point in computing the tightness of LTW in respect to DTW.

In addition to classification accuracy, computational complexity is a crucial decision criterion for choosing an appropriate time series distance measure. Our theoretical and empirical evaluation demonstrate that LTW is linear in respect to time series length, and furthermore performs pairwise distance calculations orders of magnitudes faster than DTW and cDTW, which exhibit quadratic complexity. Although cDTW in combination with lower bounding leads to quasi-linear complexity for querying large time series databases [17, 19], we have shown that this approach causes comparatively high ACCs for 1NN classification, where a multitude of unlabeled time series is compared against a set of unordered prototypes. Our empirical results on average amortized computational cost (ACC) demonstrate that our straightforward LTW approach without pruning is advantageous to 1NN classification for almost all examined time series datasets.

5. CONCLUSION AND FUTURE WORK

To speed up 1NN classification of time series, we proposed the Lucky Time Warping (LTW) distance, which determines the warping path in a greedy manner. In this work we showed that LTW:

- calculates the similarity between a pair of time series in linear time and space,
- trades classification accuracy against computational cost reasonably well,
- is able to cope with time series that exhibit warping invariance,
- is straightforward and easy to implement,
- is unaffected by warping constraints,
- does not depend on the pruning power of lower bounding techniques;

making LTW an excellent choice for 1NN classification of arbitrary time series.

Since this work only discusses a naive implementation of the LTW algorithm, there is potential for further improvements in speed, for instance by adopting early abandoning and online normalization techniques [17]. Further improvements in classification accuracy might be achieved by learning a warping path that minimizes the classification problem, and not the time series distance function.

6. REFERENCES

[1] G. E. A. P. A. Batista, X. Wang, and E. J. Keogh. A complexity-invariant distance measure for time series. In *SDM*, pages 699–710, 2011.

[2] D. J. Berndt and J. Clifford. Using dynamic time warping to find patterns in time series. In *KDD*, pages 359–370, 1994.

[3] D. J. Berndt and J. Clifford. Advances in knowledge discovery and data mining. pages 229–248. American Association for Artificial Intelligence, Menlo Park, CA, USA, 1996.

[4] M. Brown and L. Rabiner. An adaptive, ordered, graph search technique for dynamic time warping for isolated word recognition. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 30(4):535 – 544, aug 1982.

[5] K. Buza, A. Nanopoulos, and L. Schmidt-Thieme. Insight: Efficient and effective instance selection for time-series classification. In *PAKDD*, pages 149–160, 2011.

[6] S. Chu, E. J. Keogh, D. Hart, and M. J. Pazzani. Iterative deepening dynamic time warping for time series. In *SDM*, 2002.

[7] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: experimental comparison of representations and distance measures. *PVLDB*, 1(2):1542–1552, 2008.

[8] F. Itakura. Minimum prediction residual principle applied to speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(1):67–72, 1975.

[9] E. J. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. *Data Min. Knowl. Discov.*, 7(4):349–371, 2003.

[10] E. J. Keogh and M. J. Pazzani. Scaling up dynamic time warping for datamining applications. In *KDD*, pages 285–289, 2000.

[11] E. J. Keogh and C. A. Ratanamahatana. Everything you know about dynamic time warping is wrong. In *KDD*, 2004.

[12] E. J. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowl. Inf. Syst.*, 7(3):358–386, 2005.

[13] E. J. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The UCR time series classification/clustering homepage, 2011.

[14] S.-W. Kim, S. Park, and W. W. Chu. Efficient processing of similarity search under time warping in sequence databases: an index-based approach. *Inf. Syst.*, 29(5):405–420, 2004.

[15] S. MacLean and G. Labahn. Elastic matching in linear time and constant space. In *DAS*, 2010.

[16] E. Pkalska, R. P. W. Duin, and P. Paclík. Prototype selection for dissimilarity-based classifiers. *Pattern Recogn.*, 39(2):189–208, Feb. 2006.

[17] T. Rakthanmanon, B. J. L. Campana, A. Mueen, G. E. A. P. A. Batista, M. B. Westover, Q. Zhu, J. Zakaria, and E. J. Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *KDD*, pages 262–270, 2012.

[18] C. A. Ratanamahatana and E. J. Keogh. Making time-series classification more accurate using learned constraints. In *SDM*, 2004.

[19] C. A. Ratanamahatana and E. J. Keogh. Three myths about dynamic time warping data mining. In *SDM*, 2005.

[20] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[21] Y. Sakurai, M. Yoshikawa, and C. Faloutsos. Ftw: fast similarity search under the time warping distance. In *PODS*, pages 326–337, 2005.

[22] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, 2007.

[23] S. Spiegel and S. Albayrak. An order-invariant time series distance measure. In *KDIR*. SciTePress Digital Library, 2012.

[24] X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *ICML*, pages 1033–1040, 2006.

[25] B.-K. Yi, H. V. Jagadish, and C. Faloutsos. Efficient retrieval of similar time sequences under time warping. In *ICDE*, pages 201–208, 1998.