

## Article

# Improved LDTW Algorithm Based on the Alternating Matrix and the Evolutionary Chain Tree

Zheng Zou <sup>1</sup>, Ming-Xing Nie <sup>2,\*</sup>, Xing-Sheng Liu <sup>3</sup> and Shi-Jian Liu <sup>3,4,\*</sup>

<sup>1</sup> College of Computer and Cyber Security, Fujian Normal University, Fuzhou 350117, China; zouzheng2018@fjnu.edu.cn

<sup>2</sup> School of Computer Science, University of South China, Hengyang 421001, China

<sup>3</sup> Fujian Provincial Key Laboratory of Big Data Mining and Applications, Fujian University of Technology, Fuzhou 350118, China; xingshengliu1997@hotmail.com

<sup>4</sup> Fujian Provincial Key Laboratory of Information Processing and Intelligent Control, Minjiang University, Fuzhou 350108, China

\* Correspondence: niemx@usc.edu.cn (M.-X.N.); liusj2003@gmail.com (S.-J.L.)

**Abstract:** Dynamic time warping under limited warping path length (LDTW) is a state-of-the-art time series similarity evaluation method. However, it suffers from high space-time complexity, which makes some large-scale series evaluations impossible. In this paper, an alternating matrix with a concise structure is proposed to replace the complex three-dimensional matrix in LDTW and reduce the high complexity. Furthermore, an evolutionary chain tree is proposed to represent the warping paths and ensure an effective retrieval of the optimal one. Experiments using the benchmark platform offered by the University of California-Riverside show that our method uses 1.33% of the space, 82.7% of the time used by LDTW on average, which proves the efficiency of the proposed method.

**Keywords:** dynamic time warping; time series; similarity evaluation; warping path; space-time complexity



**Citation:** Zou, Z.; Nie, M.-X.; Liu, X.-S.; Liu, S.-J. Improved LDTW Algorithm Based on the Alternating Matrix and the Evolutionary Chain Tree. *Sensors* **2022**, *22*, 5305. <https://doi.org/10.3390/s22145305>

Academic Editors: Chihyu Hsu and Shuo-Tsung Chen

Received: 23 June 2022

Accepted: 14 July 2022

Published: 15 July 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

As a common data type, time series is a sequence of discrete data obtained from a target with a fixed frequency in a period. A fundamental task regarding the time series is to measure the similarity between two given ones, which is critical to downstream works in terms of classification [1–5], clustering [6–10] and pattern recognition [11–14]. The dynamic time warping (DTW) [15] algorithm and its variants [16–18] are competent in similarity evaluation [19].

Given series  $X$  and  $Y$ , if they are of the same length  $N$ , then the similarity  $S$  could be described as Expression (1).

$$S = \sum_{i=1}^N \|x_i - y_i\| \quad (1)$$

where  $\|\bullet\|$  stands for the Euclidean distance,  $x_i$  and  $y_i$  are the  $i$ th node of  $X$  and  $Y$ , respectively. However, more generally, the length of  $X$  and  $Y$  may not be the same. A key feature of DTW is that it can deal with two series of different lengths.

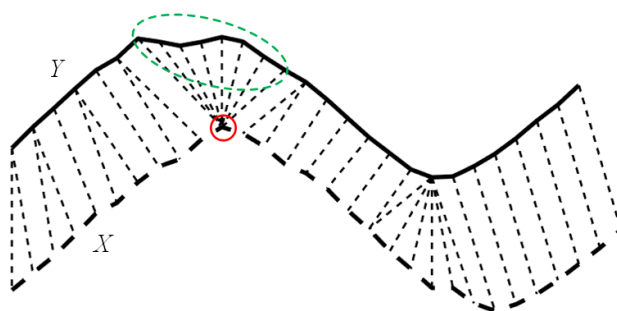
Let  $N$  and  $M$  be the length of  $X$  and  $Y$ , respectively; DTW finds the similarity by maintaining a two-dimensional cumulative distance matrix (CDM)  $D$  as shown in Expression (2). The algorithm calculates each element of  $D$  in row-major order (i.e., from left to right, from top to bottom), which starts from  $d_{1,1}$  till  $d_{N,M}$  according to Expression (3).

$$D = \begin{bmatrix} d_{1,1} & & \cdots & & d_{1,M} \\ & \ddots & & & \\ \vdots & & d_{i-1,j-1} & d_{i-1,j} & \vdots \\ & & d_{i,j-1} & d_{i,j} & \\ d_{N,1} & & \cdots & & d_{N,M} \end{bmatrix} \quad (2)$$

$$d_{i,j} = \text{dis}(x_i, y_j) + \min(d_{i-1,j}, d_{i,j-1}, d_{i-1,j-1}) \quad (3)$$

where  $\text{dis}(\bullet)$  is the distance between two nodes. After the traversal,  $d_{N,M}$  will hold the value of the similarity. The matching results (or the optimal warping path in other words) could be determined according to the CDM.

For the evaluation of series with different lengths, as depicted in Figure 1, DTW aims to find the optimal alignment between  $X$  and  $Y$  [20], and a node in  $X$  may be matched with multiple nodes in  $Y$  (and vice versa). However, if too many nodes (marked within a green dotted circle in Figure 1) are matched with the same one (marked within a red solid circle in Figure 1) which is unreasonable in a real case, it is referred to as the well-known pathological alignment problem of DTW.



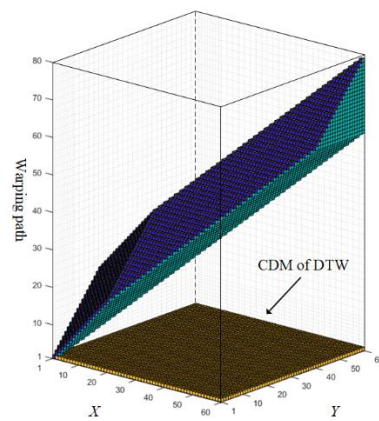
**Figure 1.** Demonstration of the pathological alignment problem of DTW, where one node in  $X$  (marked with the red solid circle) is matched with too many nodes in  $Y$  (marked within the green dotted circle).

To solve that, Zhang et al. [21] presented a state-of-the-art method named dynamic time warping under limited warping path length (LDTW). By limiting the length of the warping path in a third dimension (see Figure 2), the pathological alignment problem could be relieved. As a result, LDTW boosts the accuracy against other variants [22–25] on the benchmark platform offered by the University of California-Riverside (UCR) [26]. However, it also leads to a much higher space-time consumption.

To reduce the complexity of LDTW, an alternating matrix whose size is much smaller than the three-dimensional CDM used in LDTW is presented, and an evolutionary tree is introduced to represent the warping paths as well. The main contributions of this paper are twofold:

- (1) A two-channel matrix with an alternating scheme is proposed for similarity calculation.
- (2) A chain tree with an evolutionary scheme is proposed to find the optimal warping path with the similarity calculation process simultaneously.

The rest of this paper is organized as follows. The preliminary is given in Section 2. Section 3 presents the proposed method. The experiment and results are shown in Section 4. Section 5 concludes the work.



**Figure 2.** Comparison of the space and calculated amount between LDTW and DTW (tested on UCR data named SyntheticControl). The biggest cube is the CDM of LDTW, while the bottom part is the CDM of DTW.

## 2. Preliminary

### 2.1. DTW

DTW is a dynamic programming algorithm for calculating the similarity of two sequences, especially those of different lengths [27]. Given time series  $X$  and  $Y$  defined by Expression (4):

$$\begin{cases} X = \{x_i | 1 \leq i \leq N\} \\ Y = \{y_j | 1 \leq j \leq M\} \end{cases} \quad (4)$$

where  $N$  and  $M$  are the lengths of  $X$  and  $Y$ , respectively. If  $P(X, Y)$  defined by Expression (5) is a warping path of  $X$  and  $Y$ , each path node  $p_t$  could be defined by a pair of nodes of  $X$  and  $Y$  as shown in Expression (6).

$$P(X, Y) = \{p_t | 1 \leq t \leq L\} \quad (5)$$

$$p_t = (x_i, y_j) (i \in [1, N], j \in [1, M]) \quad (6)$$

In addition, the warping path also abides by the following restrains.

- (1)  $p_1 = (x_1, y_1), p_L = (x_N, y_M)$ ;
- (2) if  $p_t = (x_i, y_j)$  and  $p_{t+1} = (x_{i'}, y_{j'})$ , then  $0 \leq i - i' \leq 1, 0 \leq j - j' \leq 1$ .

Let  $A^{X, Y}$  denote all the warping paths of  $X$  and  $Y$ , DTW aims to find an optimal one  $P_O(X, Y)$  that possesses minimum cumulative distance as shown in Expression (7).

$$\min_{P(X, Y) \in A^{X, Y}} \sum_{(x_i, y_j) \in P(X, Y)} dis(x_i, y_j) \quad (7)$$

where  $dis(x_i, y_j)$  is the distance between two nodes  $x_i$  and  $y_j$  among a warping path  $P(X, Y)$ .

The problem could be solved in a dynamic programming way. Namely, let  $X_s^e$  (or  $Y_s^e$ ) denote the subset of  $X$  (or  $Y$ ) that starts from the  $s$ th node to the  $e$ th node, the cumulative distance of  $P_O(X_i^1, Y_j^1)$  consists of the node distance  $dis(x_i, y_j)$  and the minimum value among  $CD[P_O(X_{i-1}^1, Y_j^1)]$ ,  $CD[P_O(X_i^1, Y_{j-1}^1)]$  and  $CD[P_O(X_{i-1}^1, Y_{j-1}^1)]$  as described in Expression (8).

$$dis(x_i, y_j) + \min \left\{ CD[P_O(X_{i-1}^1, Y_j^1)], CD[P_O(X_i^1, Y_{j-1}^1)], CD[P_O(X_{i-1}^1, Y_{j-1}^1)] \right\} \quad (8)$$

where  $CD[\bullet]$  indicates the cumulative distance of a path. This is the reason for DTW to maintain the CDM and calculate according to Expression (3), which is another version of Expression (8).

## 2.2. LDTW

To ease the pathological alignment problem, besides the series length, LDTW takes the warping path length into consideration as well, which extends the original two-dimensional CDM of size  $N \times M$  to a three-dimensional matrix of size  $N \times M \times L_{UB}$ , where  $N$  and  $M$  are the lengths of two series,  $L_{UB}$  is the upper bound of the warping path length, the range of which is  $[\max(N, M) + 1, N + M - 2]$  under the rule of DTW (see Ref. [21] for the details about  $L_{UB}$ ). For example, Figure 2 showed a case that applies LDTW on UCR data named SyntheticControl, where  $N = M = 60$ ,  $L_{UB} = 79$ . The space used by LDTW is a cubic matrix of size  $60 \times 60 \times 79$ . By contrast, DTW only uses the bottom of the cube. The elements that participated in the calculation are colored in the figure as well, which is 18490 in total for LDTW and 3600 for DTW. It shows that, compared to DTW, the time and space complexity of LDTW is greatly increased.

In this paper, a matrix of size  $2 \times M \times L_{UB}$  is used to replace the above three-dimensional CDM with an alternating scheme, which reduces the cost of time and space dramatically.

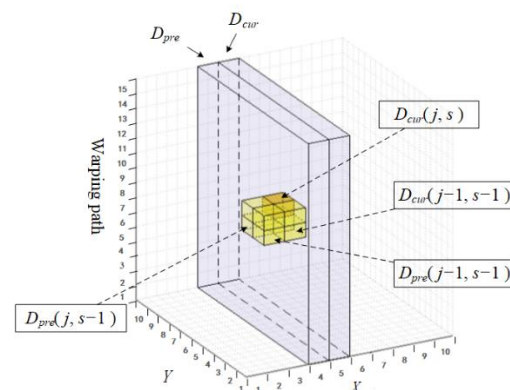
## 3. The Proposed Method

There are two goals for DTW and the variant algorithms in general, which are finding (1) the similarity and (2) the optimal warping path of two given time series. This section will present our solutions, respectively.

### 3.1. The Alternating Matrix Based Similarity Calculation

The primary innovation of the proposed method is the usage of a two-channel matrix with an alternating scheme, which can replace the three-dimensional CDM of LDTW and save a lot of computer memory.

As illustrated in Figure 3, the proposed matrix has two channels indicated by  $D_{pre}$  and  $D_{cur}$ , respectively. It could be seen as a subset of the three-dimensional CDM and travels over the CDM space during the similarity calculation process step by step. In each step, data in  $D_{pre}$  stand for the calculated result of the previous step. Moreover, it is reserved to participate in the calculation of the current step, which happens in  $D_{cur}$ . The last thing to accomplish in each step is to alternate the role of the two channels, in other words  $D_{cur}$  (or  $D_{pre}$ ) in Step  $i$  will be  $D_{pre}$  (or  $D_{cur}$ ) in Step  $i + 1$ , which is the main reason why we call our matrix the alternating matrix (AM).



**Figure 3.** The proposed two-channel alternating matrix within the CDM space.

The calculation workflow can be seen in Figure 4. The system takes the above-mentioned  $X, N, Y, M, L_{UB}$  as input and outputs the similarity  $S$  which equals to a specific element of the AM (i.e.,  $\min(\{D_{pre}[M][s] | \min S \leq s \leq L_{UB}\})$ ). The core step is the update of the AM, which is described in Algorithm 1. In the beginning, the algorithm travels over  $Y$  and the warping path dimension as shown from Step 1 to Step 4, where  $\min S$  and  $\max S$  are the ranges calculated by functions named  $\text{MinStep}()$  and  $\text{MaxStep}()$ , respectively. Readers

can find the calculation details in Ref. [21]. Step 5 specifies how an element  $D_{cur}(j, s)$ , as shown in Figure 3, is determined by pre-calculated  $D_{pre}(j, s - 1)$ ,  $D_{pre}(j - 1, s - 1)$  and  $D_{cur}(j - 1, s - 1)$ . Channel  $D_{pre}$  will be reset in Step 9 before the alternating process, for it will become  $D_{cur}$  in the next round of iteration. The iteration stops when  $i$  becomes larger than  $N$ .

---

**Algorithm 1: AM Update**


---

Input:  $X, Y, N, M, D, i, cur, pre, L_{UB}$

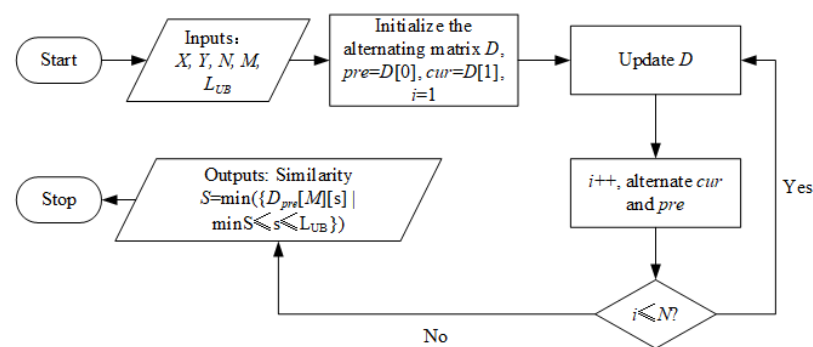
Output: updated  $D$

```

1  for  $j$  from 1 to  $M$  do
2       $minS \leftarrow \text{MinStep}(i, j)$ ,  $maxS \leftarrow \text{MaxStep}(i, j, N, M, L_{UB})$ 
3      if  $minS < maxS$  do
4          for  $s$  from  $minS$  to  $maxS$  do
5               $D_{cur}[j][s] \leftarrow \min\{D_{pre}[j][s - 1], D_{cur}[j - 1][s - 1], D_{pre}[j - 1][s - 1]\} + dis(x_i, y_j)$ 
6          end for
7      end if
8  end for
9  reset  $D_{pre}$ 

```

---



**Figure 4.** The workflow of the proposed similarity calculation process.

### 3.2. The Evolutionary Chain Tree Based Optimal Warping Path Determination

Besides the similarity, we can also find the corresponding warping path, which shows the matching pairs of two series. To achieve that, a chain tree with an evolutionary scheme is proposed. We also modified the structure of the AM, where each element possesses not only a value but also a pointer.

For example, the nodes and links of the chain tree are shown as dots and arrows in Figure 5, and six AM elements are drawn as cubes. Each cube is divided into two parts, the top part is the pointer domain leading to a corresponding tree node, while the bottom part is the value domain for the storage of the cumulative distance.

The above tree is referred to as the evolutionary chain tree (ECT) because we use a chain tree to represent the warping paths and the tree is growing and pruning dynamically during the process. The usage of ECT is another major contribution of this work.

With the ECT, the workflow demonstrated in Figure 4 can be extended to an updated version shown in Figure 6. The main differences are marked as blocks in grey, which include the growing and pruning of the ECT, and the retrieval of the optimal warping path.

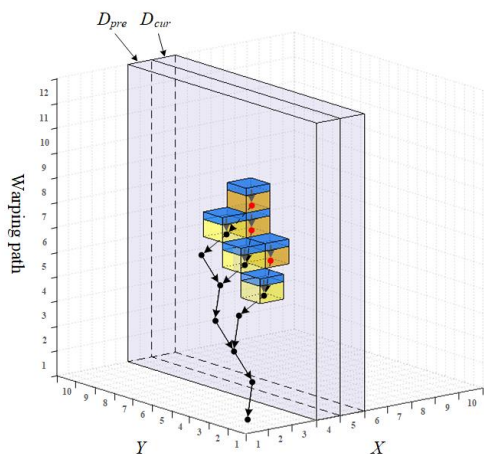


Figure 5. Illustration of the AM and ECT.

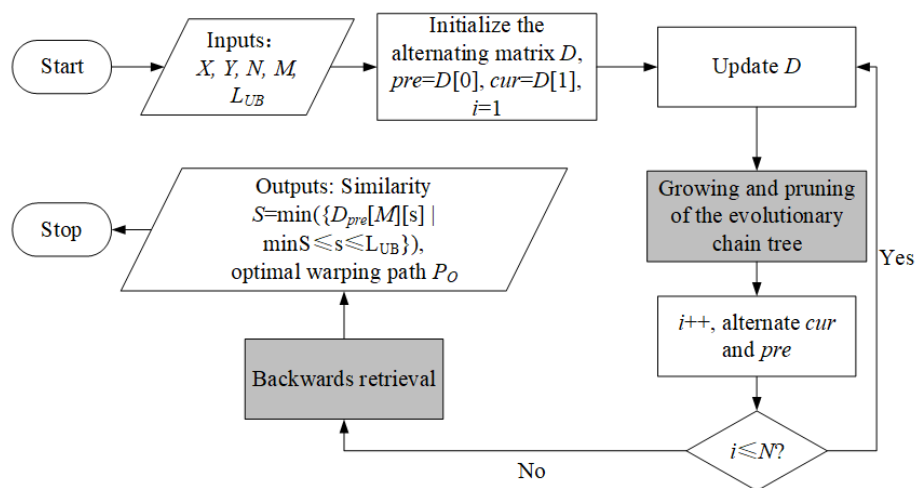


Figure 6. The updated workflow of the proposed method.

### 3.2.1. Growing

The scale of ECT grows after each update step of AM. Specifically, as soon as the computation in  $D_{cur}$  finished, tree nodes will be created and linked to the ECT. Each tree node is initialized as a structure  $p$  shown in Expression (9).

$$p : (prior = null, data = 0b0000) \tag{9}$$

where  $prior$  is the pointer that leads to a prior tree node. Description of  $data$  will be given later.

If a node  $p_s(cur, j)$  is initialized and linked from AM element  $D_{cur}(j, s)$  as shown in Figure 7a, the next question is which node is its precursor. According to Step 5 in Algorithm 1,  $D_{cur}(j, s)$  is partially determined by the minimum among  $D_{pre}(j, s - 1)$ ,  $D_{pre}(j - 1, s - 1)$  and  $D_{cur}(j - 1, s - 1)$ . Therefore, the precursor of  $p_s(cur, j)$  is the tree node that links from the minimum among  $D_{pre}(j, s - 1)$ ,  $D_{pre}(j - 1, s - 1)$  and  $D_{cur}(j - 1, s - 1)$  as well. The above processes are shown in Algorithm 2, from Steps 5 to Step 7.

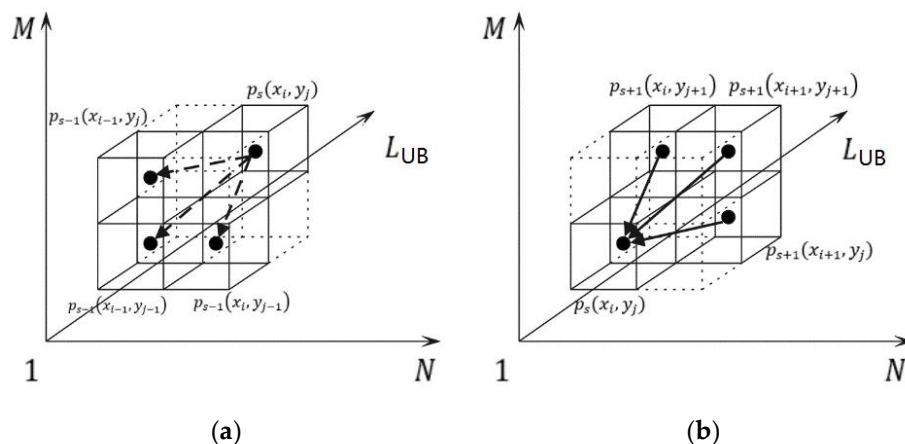


Figure 7. The potential precursor (a) and successors (b) of a tree node  $p_s(x_i, y_j)$ .

The *data* term of a tree node  $p$  is a four-digit value. The higher two digits are defined in Table 1, which is a clue to finding all the X and Y indexes of the optimal warping path nodes since we did not save them. Specifically, when retrieving the optimal warping path, it begins from the tree node linked from  $\min(\{D_{pre}[M][s] | \min S \leq s \leq L_{UB}\})$  backwards to the first one following the pointers. Because the indexes of the last node are known, with the higher two digits, it is easy to find the indexes of the rests. While the lower two digits stand for the number of its successors, which is no more than three as shown in Figure 7b. The lower two digits are crucial to the pruning process introduced in the next section. Step 8 in Algorithm 2 describes the process related to the *data* term accordingly.

**Algorithm 2:** ECT Growing

Input:  $N, M, D, i, cur, pre, L_{UB}$

Output: updated  $D$

```

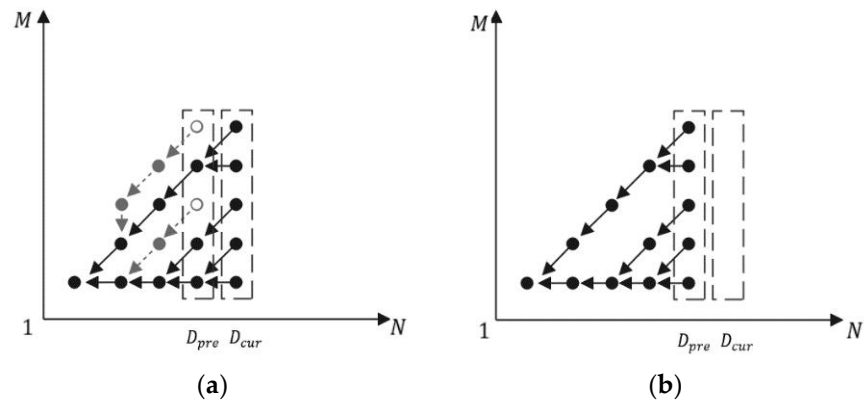
1  for j from 1 to M do
2      minS ← MinStep(i, j), maxS ← MaxStep(i, j, N, M, LUB)
3      if minS < maxS do
4          for s from minS to maxS do
5              initialize  $p : (prior = null, data = 0b0000), D_{cur}[j][s].ptr \leftarrow p$ 
6               $q \leftarrow \min\{D_{pre}[j][s-1], D_{pre}[j-1][s-1], D_{cur}[j-1][s-1]\}$ 
7               $p.prior \leftarrow q.ptr$ 
8               $p.data \leftarrow \begin{cases} 0b0100, & \text{if } q = D_{pre}[j][s-1] \\ 0b1000, & \text{if } q = D_{cur}[j-1][s-1], \quad p.prior.data ++ \\ 0b1100, & \text{if } q = D_{pre}[j-1][s-1] \end{cases}$ 
9          end for
10     end if
11 end for
    
```

Table 1. Definition of the higher two-digit data term for tree node  $p_s(cur, j)$ .

	$p_{s-1}(pre, j)$	$p_{s-1}(cur, j)$	$p_{s-1}(pre, j-1)$
$p_s(cur, j)$	0b01	0b10	0b11

3.2.2. Pruning

As the ECT grows, some branches lose their activity. Figure 8a demonstrates such a case, where two branches are not growing after new nodes have been added to ECT. Those branches can be pruned to save memory; the pruning result is shown in Figure 8b.



**Figure 8.** Illustration of ECT before (a) and after (b) pruning.

In our method, the pruning starts from leaf nodes drawn as circles in Figure 8a. They can be found from  $D_{pre}$  as shown in Algorithm 3, Step 5. If their lower two-digit data term equals 0b00, then they need to be removed because it means they have no successor.

---

**Algorithm 3:** ECT Pruning

---

Input:  $N, M, D, i, cur, pre, L_{UB}$

Output: updated  $D$

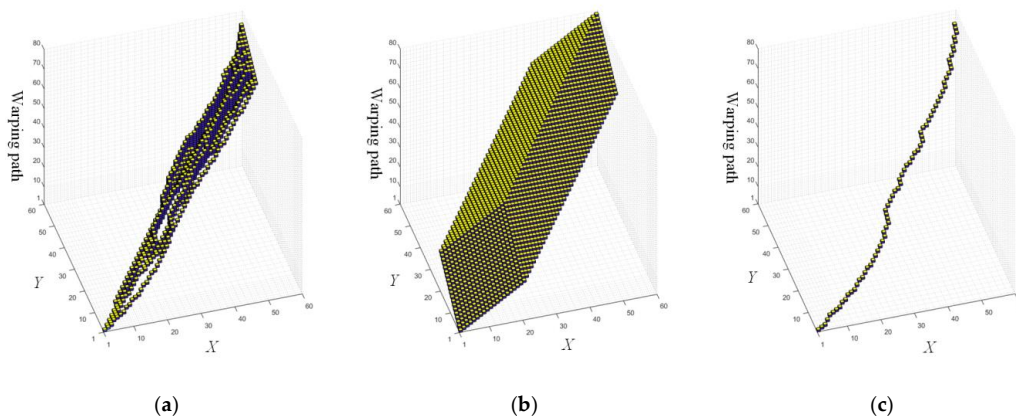
```

1  for j from 1 to M do
2      minS ← MinStep(i-1, j), maxS ← MaxStep(i-1, j, N, M, LUB)
3      if minS < maxS do
4          for s from minS to maxS do
5              p ← Dpre[j][s].ptr
6              while lower(p.data) equal to 0b00 do
7                  q ← p, p ← p.prior, p.data--, delete q
8              end while
9          end for
10     end if
11 end for

```

---

Figure 9a shows the final ECT applying the proposed method on SyntheticControl. Moreover, if no pruning is used, it would look like the one shown in Figure 9b. Figure 9c shows the optimal warping path.



**Figure 9.** The final ECT (a) with pruning and (b) without pruning. (c) The optimal warping path that extracted from the final ECT.



#### 4. Experiments and Results

The proposed method was implemented using the C++ programming language. The public dataset UCR [26] was adopted for the 1-NN classification tests on a desktop computer with AMD Ryzen 7 5800X 3.80 GHz CPU, 64 GB memory. We compared our method with LDTW in terms of time and space consumption.

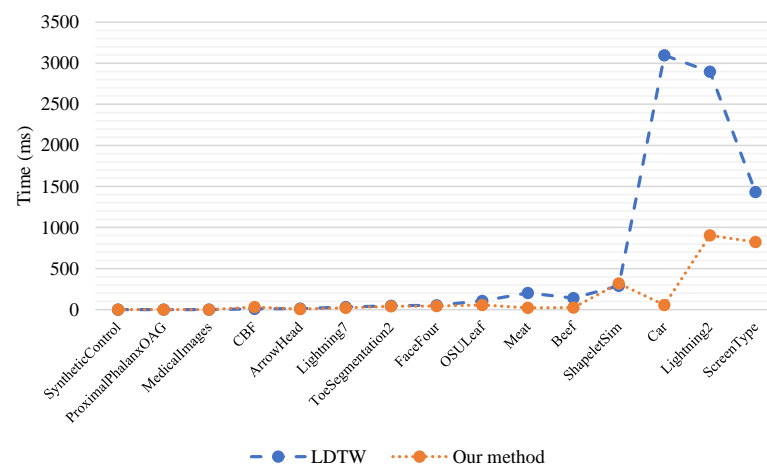
##### 4.1. Comparisons

To compare our method with LDTW in space costs, we tested it on all species in UCR. We selected the result of 15 data points for showing, and each is of a different name and length as described in the first and second columns of Table 2. There are two key phases in our method, namely the similarity calculation phase and optimal warping path determination phase, therefore we recorded the space cost of them as Ph1(MB) and Ph2(MB). As the table shows, our method uses 1.33% of the space used by LDTW on average.

**Table 2.** Comparisons between LDTW and our method in terms of space costs.

Data Name	Length	LDTW(MB)	Our Method		
			Ph1 (MB)	Ph2 (MB)	Total (MB)
SyntheticControl	60	1.02	0.03	0.01	0.04
ProximalPhalanxOAG	80	1.95	0.05	0.00	0.05
MedicalImages	99	3.89	0.08	0.01	0.09
CBF	128	12.75	0.20	0.25	0.45
ArrowHead	251	60.32	0.48	0.00	0.48
Lightning7	319	132.76	0.83	0.13	0.96
ToeSegmentation2	343	169.64	0.99	0.31	1.30
FaceFour	350	180.85	1.03	0.25	1.29
OSULeaf	427	321.33	1.51	0.47	1.98
Meat	448	343.00	1.53	0.00	1.53
Beef	470	400.27	1.70	0.02	1.72
ShapeletSim	500	567.44	2.27	0.91	3.18
Car	577	755.66	2.62	0.12	2.74
Lightning2	637	1211.99	3.81	4.40	8.21
ScreenType	720	1657.18	4.60	15.39	19.99

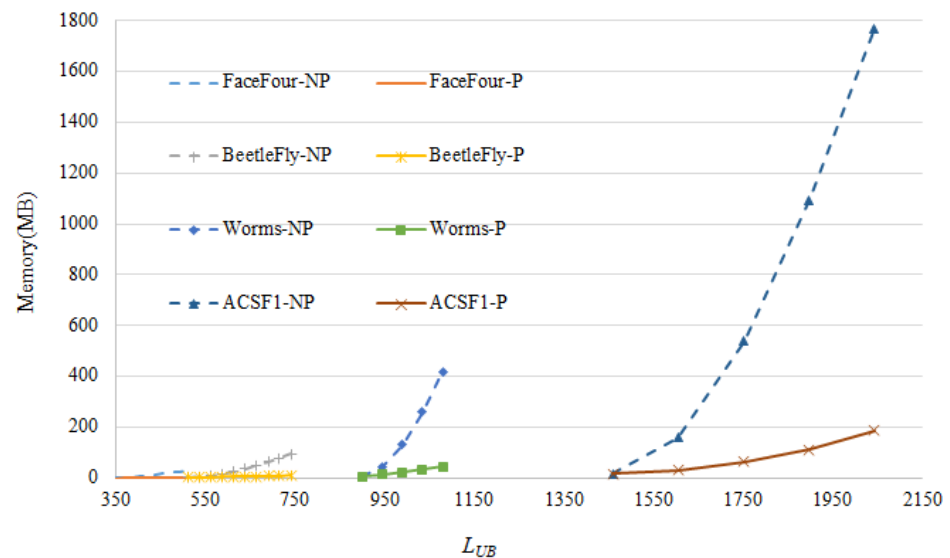
Our comparison was also completed in time costs. According to the results shown in Figure 10, there are 15 data points which are organized in ascending order of scale in the first column of Table 2 in the horizontal direction, and there are the specific time costs (ms) of our method and LDTW in the vertical direction. As the scale of the time series increases from left to right, the superiority of our method becomes more obvious.



**Figure 10.** Comparisons between LDTW and our method in terms of time costs.

#### 4.2. Ablation Experiment

To show the contribution of the pruning proposed in our method, system performance with and without pruning is investigated. As Figure 11 shows, the space consumption could be greatly reduced with the pruning process. In addition, it is normal that the space cost rises along with the increase of parameter  $L_{UB}$ . With the help of pruning, few variations have been found in Figure 11, compared to the case without pruning which is sensitive to the choice of  $L_{UB}$ . The scales of the data used in Figure 11 are listed in Table 3.



**Figure 11.** The ablation experiment results. The horizontal axis is the parameter  $L_{UB}$ , the vertical axis is the memory cost running on different data. \*-P and \*-NP stand for the method with and without pruning, respectively.

**Table 3.** The scales of the data shown in Figure 11.

Data	FaceFour	BeetleFly	Worms	ACSF1
Scale	$350 \times 350$	$512 \times 512$	$900 \times 900$	$1460 \times 1460$

## 5. Discussion

Thanks to the proposed alternating matrix, great achievement has been made in reducing the memory cost compared to the LDTW method. The price of this huge deflation is the need for an additional data structure to maintain the warping paths, as well as a new strategy for optimal warping path retrieval. We solve that problem by the proposed evolutionary chain tree, which will sacrifice little time and space, but it is just a drop in the ocean compared to the contributions. The performance of the proposed method still outranges the LDTW a lot.

Another issue is about the choice of  $L_{UB}$ , which is the only parameter in this method. The usage and setting criteria of  $L_{UB}$  in our work follow the idea introduced by the LDTW algorithm [21]. In experiments, we found that different values of  $L_{UB}$  may slightly alter the accuracy, but it is insensitive to our final space costs as shown in the ablation experiment. Therefore, to get a fairer comparison, we adopted the same method as [21] for  $L_{UB}$  to keep a similar parameters environment.

## 6. Conclusions

This paper proposes a novel resolution for recording and exploding wrapping paths with much less space-time complexity. Firstly, a two-channel matrix is created and travels over the entire cumulative distance space with an alternating scheme to calculate the

similarity. Secondly, a chain tree is involved to record all warping paths, and the tree is gradually growing and pruned along with the matrix alternating simultaneously, which ensures an efficient retrieval of the optimal path. Experiments running on the UCR benchmark show that our method uses 1.33% of the space, 82.7% of the time used by LDTW on average. Future work would focus on improving the evaluation accuracy.

**Author Contributions:** Manuscript writing, Z.Z.; experiments, X.-S.L.; project administration, S.-J.L.; funding acquisition and proofreading, M.-X.N. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the Natural Science Foundation of Hunan Province (grant number 2021JJ30574), the Research Foundation of Education Bureau of Hunan Province (grant number 21B0424), the Natural Science Foundation of Fujian Province (grant number JAT210283, 2022J01932), the Open Project of Fujian Provincial Key Laboratory of Information Processing and Intelligent Control (grant number MJUKF-IPIC202208).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data presented in this study are available on request from the corresponding author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Lichtenauer, J.F.; Hendriks, E.A.; Reinders, M.J. Sign language recognition by combining statistical DTW and independent classification. *IEEE Trans. Pattern Anal.* **2008**, *30*, 2040–2046. [[CrossRef](#)]
2. Zhang, Z.; Tang, P.; Hu, C.; Liu, Z.; Zhang, W.; Tang, L. Seeded Classification of Satellite Image Time Series with Lower-Bounded Dynamic Time Warping. *Remote Sens.* **2022**, *14*, 2778. [[CrossRef](#)]
3. Amerineni, R.; Gupta, L.; Steadman, N.; Annauth, K.; Burr, C.; Wilson, S.; Barnaghi, P.; Vaidyanathan, R. Fusion Models for Generalized Classification of Multi-Axial Human Movement: Validation in Sport Performance. *Sensors* **2021**, *21*, 8409. [[CrossRef](#)] [[PubMed](#)]
4. Li, J.; Zhang, H.; Dong, Y.; Zuo, T.; Xu, D. An Improved Self-Training Method for Positive Unlabeled Time Series Classification Using DTW Barycenter Averaging. *Sensors* **2021**, *21*, 7414. [[CrossRef](#)]
5. Lei, T.C.; Wan, S.; Wu, Y.C.; Wang, H.-P.; Hsieh, C.-W. Multi-Temporal Data Fusion in MS and SAR Images Using the Dynamic Time Warping Method for Paddy Rice Classification. *Agriculture* **2022**, *12*, 77. [[CrossRef](#)]
6. Kumar, D.; Wu, H.; Rajasegarar, S.; Leckie, C.; Krishnaswamy, S.; Palaniswami, M. Fast and scalable big data trajectory clustering for understanding urban mobility. *IEEE Trans. Intell. Transp.* **2018**, *19*, 3709–3722. [[CrossRef](#)]
7. Petitjean, F.; Ketterlin, A.; Gançarski, P. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recogn.* **2011**, *44*, 678–693. [[CrossRef](#)]
8. Jiang, Y.; Qi, Y.; Wang, W.K.; Bent, B.; Avram, R.; Olgin, J.; Dunn, J. EventDTW: An Improved Dynamic Time Warping Algorithm for Aligning Biomedical Signals of Nonuniform Sampling Frequencies. *Sensors* **2020**, *20*, 2700. [[CrossRef](#)]
9. He, Y.; Zhang, X.; Wang, R.; Cheng, M.; Gao, Z.; Zhang, Z.; Yu, W. Faulty Section Location Method Based on Dynamic Time Warping Distance in a Resonant Grounding System. *Energies* **2022**, *15*, 4923. [[CrossRef](#)]
10. Debella, T.T.; Shawel, B.S.; Devanne, M.; Weber, J.; Woldegebreal, D.H.; Pollin, S.; Forestier, G. Deep Representation Learning for Cluster-Level Time Series Forecasting. *Eng. Proc.* **2022**, *18*, 22.
11. Cui, J.-W.; Li, Z.-G.; Du, H.; Yan, B.-Y.; Lu, P.-D. Recognition of Upper Limb Action Intention Based on IMU. *Sensors* **2022**, *22*, 1954. [[CrossRef](#)]
12. Zhao, S.; Cai, H.; Li, W.; Liu, Y.; Liu, C. Hand Gesture Recognition on a Resource-Limited Interactive Wristband. *Sensors* **2021**, *21*, 5713. [[CrossRef](#)]
13. Li, T.; Shi, C.; Li, P.; Chen, P. A Novel Gesture Recognition System Based on CSI Extracted from a Smartphone with Nexmon Firmware. *Sensors* **2021**, *21*, 222. [[CrossRef](#)]
14. Li, H.; Khoo, S.; Yap, H.J. Implementation of Sequence-Based Classification Methods for Motion Assessment and Recognition in a Traditional Chinese Sport (Baduanjin). *Int. J. Environ. Res. Public Health* **2022**, *19*, 1744. [[CrossRef](#)]
15. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the Knowledge Discovery and Data Mining Workshop, Seattle, WA, USA, 31 July 1994.
16. Phan, T.T.H.; Caillaud, E.P.; Lefebvre, A.; Bigand, A. Dynamic time warping-based imputation for univariate time series data. *Pattern Recogn. Lett.* **2020**, *139*, 139–147. [[CrossRef](#)]
17. Guo, F.; Zou, F.; Luo, S.; Liao, L.; Wu, J.; Yu, X.; Zhang, C. The Fast Detection of Abnormal ETC Data Based on an Improved DTW Algorithm. *Electronics* **2022**, *11*, 1981. [[CrossRef](#)]

18. Chang, C.; Shaw, T.; Goutam, A.; Lau, C.; Shan, M.; Tsai, T.J. Parameter-Free Ordered Partial Match Alignment with Hidden State Time Warping. *Appl. Sci.* **2022**, *12*, 3783. [[CrossRef](#)]
19. Gong, L.; Chen, B.; Xu, W.; Liu, C.; Li, X.; Zhao, Z.; Zhao, L. Motion Similarity Evaluation between Human and a Tri-Co Robot during Real-Time Imitation with a Trajectory Dynamic Time Warping Model. *Sensors* **2022**, *22*, 1968. [[CrossRef](#)]
20. Combes, F.; Fraiman, R.; Ghattas, B. Time Series Sampling. *Eng. Proc.* **2022**, *18*, 32.
21. Zhang, Z.; Tavenard, R.; Bailly, A.; Tang, X.; Tang, P.; Corpetti, T. Dynamic time warping under limited warping path length. *Inform. Sciences* **2017**, *393*, 91–107.
22. Sakoe, H.; Chiba, S. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [[CrossRef](#)]
23. Anantasech, P.; Ratanamahatana, C.A. Enhanced weighted dynamic time warping for time series classification. In Proceedings of the Third International Congress on Information and Communication Technology, London, UK, 27–28 February 2018.
24. Jeong, Y.S.; Jeong, M.K.; Omitaomu, O.A. Weighted dynamic time warping for time series classification. *Pattern Recogn.* **2011**, *44*, 2231–2240. [[CrossRef](#)]
25. Ratanamahatana, C.A.; Keogh, E. Making time-series classification more accurate using learned constraints. In Proceedings of the 2004 SIAM International Conference on Data Mining, Lake Buena Vista, FL, USA, 22–24 April 2004.
26. Dau, H.A.; Bagnall, A.; Kamgar, K.; Yeh, C.C.M.; Zhu, Y.; Gharghabi, S.; Ratanamahatana, C.A.; Keogh, E. The UCR time series archive. *IEEE/CAA J. Automatic.* **2019**, *6*, 1293–1305. [[CrossRef](#)]
27. Cao, Y.; Ma, S.; Cao, Y.; Pan, G.; Huang, Q.; Cao, Y. Similarity Evaluation Rule and Motion Posture Optimization for a Manta Ray Robot. *J. Mar. Sci. Eng.* **2022**, *10*, 908. [[CrossRef](#)]