A Minimum-First Algorithm for Dynamic Time Warping on Time Series*

Bo-Xian Chen[1], Kuo-Tsung Tseng[2], and Chang-Biau Yang[1†]

[1] National Sun Yat-sen University, Kaohsiung, Taiwan
[†]cbyang@cse.nsysu.edu.tw (Corresponding author)
[2] National Kaohsiung University of Science and Technology, Kaohsiung, Taiwan

Abstract. *In the* time series classification *(TSC) problem, the calculation of the distance of two time series is the kernel issue. One of the famous methods for the distance calculation is the* dynamic time warping *(DTW) with $O(n^2)$ time complexity, based on the dynamic programming. It takes very long time when the data size is large. In order to overcome the time consuming problem, the* dynamic time warping with window *(DTWW) combines the warping window into DTW calculation. This method reduces the computation time by restricting the number of possible solutions, so the answer of DTWW may not be the optimal solution. In this paper, we propose the minimum-first DTW method (MDTW) that expands the possible solutions in the minimum first order. Our method not only reduces the required computation time, but also gets the optimal answer.*

Keywords: time series classification · dynamic time warping · dynamic programming · minimum first order.

1 Introduction

The *time series classification* (TSC) problem [9, 12, 13] is to determine which category the given time series belongs to. The TSC problem can usually be applied in several fields, such as word recognition [2], gesture recognition [6], robotics [4], finance, and biometrics [14].

There are several measurement skills for solving the TSC problem, such as the distance methods [11, 18], and the shapelet methods [7, 17]. Taking the distance method for instance, the idea of *Euclidean distance* (ED) is direct and simple [5]. However, it cannot solve the time distortion problem. the *dynamic time warping* (DTW) [3] overcomes the time distortion problem and reports the optimal solution. It is a pity that the time complexity of DTW is $O(n^2)$. When the lengths of input time series are large, the distance calculation with DTW needs much time.

To improve the computation efficiency of DTW, many *dynamic time warping with window* (DTWW) methods [8, 15, 16] have been presented in the past years.

* This research work was partially supported by the Ministry of Science and Technology of Taiwan under contract MOST 107-2221-E-110-033.

The main concept of these methods is to avoid some terrible alignments and then to reduce the unnecessary computations. All these DTWW methods utilize predefined windows and they face the same issue that the result may not be optimal since some computations are omitted. We then present the minimum-first DTW method (MDTW) with an adaptive window which calculates the minimum first and stops if and only if the optimal result is obtained.

The rest of this paper is organized as follows. Section 2 presents some well-known DTWW methods. We present our MDTW method in Section 3. Section 4 shows the experimental results that state the efficiency of our method. Finally, Section 5 concludes the paper and provides some advices for future works.

2 Dynamic Time Warping with Window

DTW is time-consuming since it requires $O(n^2)$ time to find the answer. Therefore, the concept of *dynamic time warping with window* (DTWW) was proposed by Itakura [8] in 1975, and Sakoe and Chiba [16] in 1978. Both of their methods belong to the global constraint. The goal of constraints is to let the warping path be closer to the diagonal and avoid the undesired path. In the method of Sakoe and Chiba, assume that the warping windows size is r. Then the path is only permitted within the width r, i.e. $|i-j| < r$. The warping path of Itakura's algorithm is bounded by two slopes S and $\frac{1}{S}$. Sakoe and Chiba used a diagonal with a fixed width.

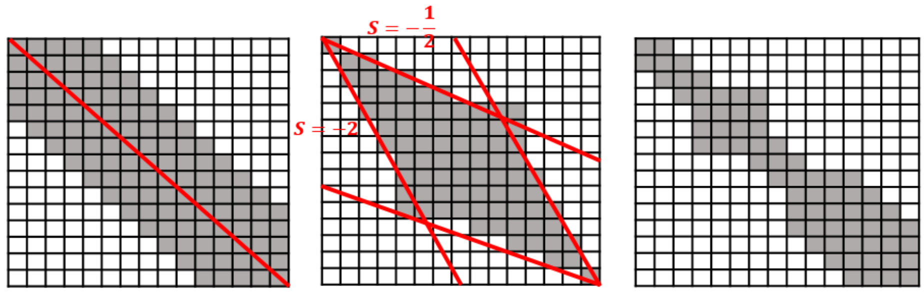
The well-known global constraints, proposed by Sakoe with Chiba, Itakura, and Ratanamahatana with Keogh are shown in Figure 1. It is worth to notice that white cells are not calculated for time-saving, so the answer may not be optimal. The spirit of our MDTW method is like DTWW methods. What is in common among MDTW and DTWW is to reduce calculated cells, but DTWW restricts the range of calculated cells with predefined windows, while MDTW changes the calculating order of cells. More specifically, MDTW calculate cells by the minimum-first order and stops if the optimal answer is obtained. Thus, the remaining cells can be ignored for time-saving.

3 Our Method

In this Section, we propose our MDTW method which is a method with an adaptive window that avoids the disadvantage of DTWW. MDTW combines the concept of dynamic time warping with an adaptive window and gets the globally optimal answer much faster.

3.1 An Example for Illustrating Our Method

We first give an example to demonstrate our concept in Figure 2. Given two time series $A = \{2, 9, 8, 8, 5, 4, 2, 1, 5\}$ and $B = \{3, 7, 4, 1, 3, 2, 1, 7\}$, we first initialize a two-dimensional matrix M to calculate the DTW distance between A and B.



(a) (b) (c)

Fig. 1: Three warping windows of DTWW. (a) Sakoe-Chiba band [16], , bounded by $|i-j| < r = 5$. (b) Itakura parallelogram [8], bounded by slopes $S = -2$ and $S = -\frac{1}{2}$. (c) Ratanamahatana-Keogh band [15].

The top-left (yellow) cell is calculated as the starting point, as shown in Figure 2a.

In the beginning, we expand three adjacent cells from cell $M[1,1]$, then we insert these expanded cells, $\langle value, row\ index, column\ index \rangle$ denoted as $\langle MV, i, j \rangle$, into a priority queue Q. Thus, we have $Q = \{\langle 3,2,2\rangle, \langle 6,1,2\rangle, \langle 7,2,1\rangle\}$, shown as the blue cells in Figure 2b. Next, we pop out the minimal cell from Q. In Figure 2b, the minimal cell $\langle MV, i, j\rangle$ in Q is $\langle 3,2,2\rangle$ (the blue circled cell). So, $M[2,2]$ (blue circled) is the next cell to be expanded, and its current cumulative distance is 3.

In Figure 2c, expanded from $M[2,2] = 3$, so get three cells $M[3,2] = 4$, $M[2,3] = 8$ and $M[3,3] = 7$. After inserting these cells into Q, and we get $Q = \{\langle 4,3,2\rangle, \langle 6,1,2\rangle, \langle 7,2,1\rangle\ \langle 7,3,3\rangle\ \langle 8,2,3\rangle\}$. This time, the minimal cell in Q is $\langle 4,3,2\rangle$ ($M[3,2] = 4$). Repeat the procedure until the bottom-right cell is expanded. In Figures 2b to 2h, the orange cells are the current minima, which are expanded to blue cells. The circled cells are the next minima and Figure 2i shows the final result.

1NN-DTW (one nearest neighbor DTW) is one of well-known methods used in the TSC problem [4, 10]. The straightforward method for 1NN-DTW is to search the database of time series one by one, and to obtain the time series having the minimum DTW distance with the query series. In the above example, the value of $M[9,8]$ is a new threshold for searching time series. If the current minimal value exceeds the threshold, we can stop the calculation of the current time series, even we do not reach the most bottom-right cell. If the distance is less than the current threshold, the threshold is updated.

3.2 The Irreplaceable Property

This section introduces the irreplaceable property of DTW. Based on this property, the value of each expanded cell cannot be updated any more.

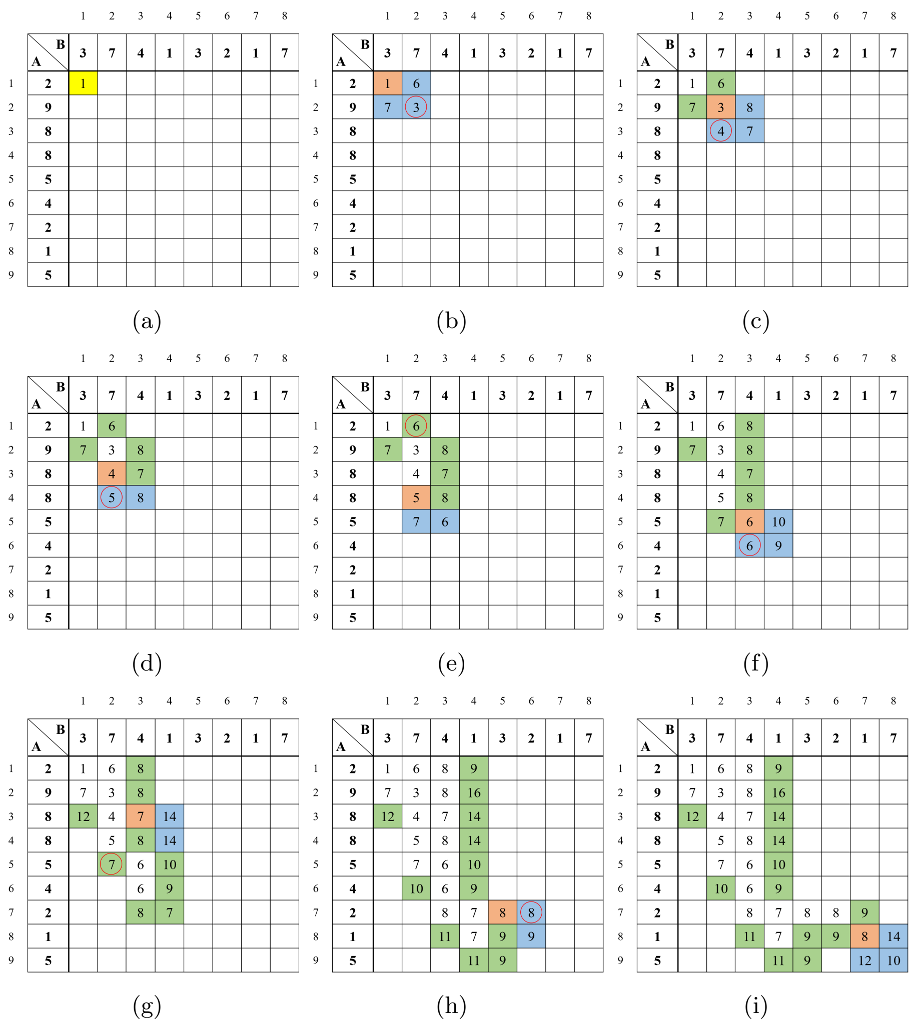


Fig. 2: The expansion steps of our MDTW with two time series $A = \{2, 9, 8, 8, 5, 4, 2, 1, 5\}$ and $B = \{3, 7, 4, 1, 3, 2, 1, 7\}$.

Theorem 1. *If the lattice M for calculating the DTW distance is expanded with the minimum first order, then the expanded cells already have their cumulative distances, and their values cannot be replaced afterwards.*

Proof. The DP formula for calculating DTW distance with M for $A = \{a_1, a_2, \cdots, a_m\}$, $B = \{b_1, b_2, \cdots, b_n\}$ is given as follows.

$$M_{i,j} = \begin{cases} 0 & \text{if } \ i = 0 \text{ and } j = 0, \\ \infty & \text{if } \ i = 0 \text{ or } j = 0, \\ & \text{and } i \neq j, \\ dis(a_i, b_j) + \min \begin{cases} M_{i-1,j} \\ M_{i,j-1} \\ M_{i-1,j-1} \end{cases} & \text{if } \ 1 \leq i \leq m \text{ and } 1 \leq j \leq n. \end{cases} \quad (1)$$

It is clear that the value of $M_{i,j}$ comes from the minimum of $M_{i-1,j}$, $M_{i,j-1}$ and $M_{i-1,j-1}$. Suppose that $M_{i,j}$ is expanded from $M_{i-1,j-1}$. In this situation, $M_{i-1,j-1}$ the minimum of $M_{i-1,j}$, $M_{i,j-1}$ and $M_{i-1,j-1}$, and $M_{i-1,j-1}$ is extracted from the queue before the other two. Accordingly, when $M_{i-1,j}$ or $M_{i,j-1}$ is extracted from the queue, $dis(a_i, b_j)+ M_{i-1,j}$ or $dis(a_i, b_j)+ M_{i,j-1}$ cannot be the answer of $M_{i,j}$. In other words, the value of $M_{i,j}$ cannot be replaced afterwards.

If $M_{i,j}$ is expanded from $M_{i,j-1}$ or $M_{i-1,j}$, it can be proved similarly.

3.3 The Minimum First Order

The pseudo code of our MDTW algorithm is shown in Algorithm 1. The threshold T is set to infinity initially, and it is updated along with the time series one by one. So, T may become lower and lower or unchanged. If we find the most similar time series, we can reduce much more calculation in later searches. Based on Theorem 1, we do not need to initialize the two-dimensional matrix, and we need only to record the expanded cells instead. We expand the cells until we get the bottom-right cell or the minimal value in queue Q exceeds T.

Algorithm 1 Minimum first DTW (MDTW)

Input: two time series A, B and threshold T
Output: *distance of* A *and* B ▷ if $MV > T$, then return null

i, $j = 1$
$MV = |a_i - b_j|$ ▷ current minimal value
$Q = \{\langle \mathrm{MV}, i, j \rangle\}$ ▷ insert unexpanded cells into queue
while $T > MV$ **do**
    if $(i+1, j)$ is not in Q and does not exceed boundary **then**
        Insert $\langle |a_{i+1} - b_j| + MV, i+1, j \rangle$ into Q
    end if
    if $(i, j+1)$ is not in Q and does not exceed boundary **then**
        Insert $\langle |a_i - b_{j+1}| + MV, i, j+1 \rangle$ into Q
    end if
    if $(i+1, j+1)$ is not in Q and does not exceed boundary **then**
        Insert $\langle |a_{i+1} - b_{j+1}| + MV, i+1, j+1 \rangle$ into Q
    end if
    if (m, n) in Q **then**
        return MV of cell (m, n)
    end if
    $\langle MV, i, j \rangle \leftarrow \min(Q)$ ▷ minimal value of MV in Q
end while

4 Experimental Results

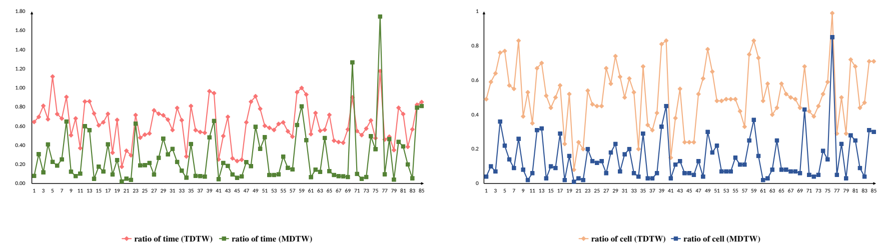
The computer environment of our experiments is Intel(R) Core(TM) i7-4790 CPU @ 3.6GHz and memory 8GB RAM. The experimental datasets come from UEA & UCR time series repository [1]. There are totally 85 classes in the repository, where each of them has its own distinct training set size, testing set size, time series length and different number of classes. Since they are open datasets, we omit the detailed description of the datasets.

To improve 1NN-DTW, the concept of threshold T can also be applied. When the DTW distance of the query series and one target series is calculated, once the distance exceeds T, we can stop the distance calculation. Here, this improvement is denoted as TDTW.

Figure 3 shows ratios of the computational time and expanded cells of TDTW and MDTW compared with the original DTW method with the threshold and our method MDTW. It is obvious that lines of MDTW are almost lower than the lines of TDTW. In other words, MDTW gets optimal answer with less computational time and fewer expanded cells. The ratio of expanded cells for MDTW are between 0.01 (the best case - Wafer) and 0.85 (the worst case - ShapeletSim), and most of them are nearly between 0.1 to 0.3. This shows that MDTW expands fewer cells to get the optimal answer. The average ratio of the computational time for MDTW is less than 0.3.

It is worth to be discussed in future that our MDTW method beats TDTW in 83 datasets, but takes more time than the original DTW and TDTW in datasets

Phoneme and ShapeletSim. Though we expand fewer cells in both cases, the computational times are longer than the original DTW method. We shall investigate the features of these two datasets, so that we may improve our MDTW method.



(a) (b)

Fig. 3: The ratios of computational time and expanded cells of TDTW and MDTW (a) Ratios of execution time. (b) Ratios of expanded cells.

5 Conclusion

In this paper, we propose the minimum-first DTW method (MDTW) for calculating the DTW distance of two time series. MDTW expands the lattice cells with the minimum-first order, The computation is like DTW with an adaptive window. MDTW finds the optimal answer and reduces the computational time. As the experiment results show, most cases (83/85) in the experimental datasets require less computational time than the original DTW, and the ratios of expanded cell that we have to calculate are between 0.1 and 0.3.

Our method performs very well on similar but distorted data, because the warping path is almost along the diagonal direction. However, the performance of our method in the dataset which falls and rises extremely is bad, because the DTW distance is large and many cells have to be expanded.

Our method saves much more time for the TSC problem. In the future, dynamic adjustment of increment in every turn may be studied, and we may try to discover more relationships between cells or the method of pruning unnecessary cells. Moreover, we may design a measure method to evaluate whether our method can perform well or not in advance.

References

1. Bagnall, A., Lines, J., Bostrom, A., Large, J., Keogh, E.: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. Data Mining and Knowledge Discovery **31**(3), 606–660 (May 2016)

2. Bailly, A., Malinowski, S., Tavenard, R., Chapel, L., Guyet, T.: Dense bag-of-temporal-sift-words for time series classification. In: Proceedings of International Workshop on Advanced Analysis and Learning on Temporal Data (AALTD). pp. 17–30. Springer, Porto, Portugal (2015)
3. Bellman, R., Kalaba, R.: On adaptive control processes. IRE Transactions on Automatic Control **4**(2), 1–9 (1959)
4. Buza, K., Nanopoulos, A., Schmidt-Thieme, L.: Time-series classification based on individualised error prediction. In: Computational Science and Engineering (CSE), 2010 IEEE 13th International Conference on. pp. 48–54. IEEE (2010)
5. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., Keogh, E.: Querying and mining of time series data: experimental comparison of representations and distance measures. Proceedings of the Very Large Data Bases Endowment **1**(2), 1542–1552 (Aug 2008)
6. Hamilton, J.D.: Time series analysis, vol. 2. Princeton university press Princeton (1994)
7. Hills, J., Lines, J., Baranauskas, E., Mapp, J., Bagnall, A.: Classification of time series by shapelet transformation. Data Mining and Knowledge Discovery **28**(4), 851–881 (July 2014)
8. Itakura, F.: Minimum prediction residual principle applied to speech recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **23**(1), 67–72 (Feb 1975)
9. Jain, B.J., Spiegel, S.: Time series classification in dissimilarity spaces. In: Proceedings of 1st International Workshop on Advanced Analytics and Learning on Temporal Data. Porto, Portugal (2015)
10. Jeong, Y.S., Jeong, M.K., Omitaomu, O.A.: Weighted dynamic time warping for time series classification. Pattern Recognition **44**(9), 2231–2240 (Sept 2011)
11. Lines, J., Bagnall, A.: Time series classification with ensembles of elastic distance measures. Data Mining and Knowledge Discovery **29**(3), 565–592 (May 2015)
12. Łuczak, M.: Univariate and multivariate time series classification with parametric integral dynamic time warping. Journal of Intelligent and Fuzzy Systems **33**(4), 2403–2413 (Sept 2017)
13. Morel, M., Achard, C., Kulpa, R., Dubuisson, S.: Time-series averaging using constrained dynamic time warping with tolerance. Pattern Recognition **74**, 77–89 (Feb 2018)
14. Parthasaradhi, S.T., Derakhshani, R., Hornak, L.A., Schuckers, S.A.: Time-series detection of perspiration as a liveness test in fingerprint devices. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **35**(3), 335–343 (July 2005)
15. Ratanamahatana, C.A., Keogh, E.: Making time-series classification more accurate using learned constraints. In: Proceedings of the 4th SIAM International Conference on Data Mining. pp. 11–22. SIAM, Florida, USA (2004)
16. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Transactions on Acoustics, Speech, and Signal Processing **26**(1), 43–49 (1978)
17. Shah, M., Grabocka, J., Schilling, N., Wistuba, M., Schmidt-Thieme, L.: Learning DTW-shapelets for time-series classification. In: Proceedings of the 3rd IKDD Conference on Data Science (CODS). p. 3. ACM, New york, USA (Mar 2016)
18. Stefan, A., Athitsos, V., Das, G.: The move-split-merge metric for time series. IEEE transactions on Knowledge and Data Engineering **25**(6), 1425–1438 (June 2013)