Segmentation of Time Series in Improving Dynamic Time Warping

Ruizhe Ma
Department of Computer Science
Georgia State University
Atlanta, USA
rma1@student.gsu.edu

Azim Ahmadzadeh
Department of Computer Science
Georgia State University
Atlanta, USA
aahmadzadeh1@student.gsu.edu

Soukaïna Filali Boubrahimi
Department of Computer Science
Georgia State University
Atlanta, USA
sfilaliboubrahimi1@student.gsu.edu

Rafal A. Angryk
Department of Computer Science
Georgia State University
Atlanta, USA
rangryk@cs.gsu.edu

***Abstract*—Since its introduction to the computer science community, the Dynamic Time Warping (DTW) algorithm has demonstrated good performance with time series data. While this elastic measure is known for its effectiveness with time series sequence comparisons, the possibility of pathological warping paths weakens the algorithms potential considerably. Techniques centering on pruning off impossible mappings or lowering data dimensions such as windowing, slope weighting, step pattern, and approximation have been proposed over the years to reduce the possibility of pathological warping paths with Dynamic Time Warping. However, because the current DTW improvement techniques are mostly global methods, they are either limited in effect or limit the warping path excessively. We believe segmenting time series at significant feature points will alleviate some of the pathological warpings, and at the same time allowing us to obtain more intuitive warpings. Our heuristic approaches the problem from the human perspective of sequence comparison: by identifying global similarity before local similarities. We use easily identifiable peaks as the significant feature. The final distance is the DTW distance sum of all segments of time series. In this paper, we explore the impact of different peak identification parameters on Dynamic Time Warping and demonstrate how segmentation can help to avoid pathological warpings.**

***Index Terms*—time series, Dynamic Time Warping, feature selection**

I. Introduction

With the development of data collection and storage, time series data is now commonly applied in a variety of domains, from voice recognition, the stock market, to solar activities, medical research, and many other scientific and engineering fields where measurements in the temporal sense are important. With more data, the need to effectively process and compare data is essential. Distance measures can be categorized as lock-step and elastic. Lock-step measures generally refer to Lp norms, meaning the *i*-th element in one sequence is always mapped to the *i*-th element in another sequence. While elastic measures allow for one-to-many, or even one-to-none mappings [1]. With the commonly seen temporal discrepancies in time series sequences, traditional lock-step measures are not as effective as elastic when identifying similarities [2].

Dynamic Time Warping (DTW) algorithm is a widely used elastic measure [3]. DTW finds the mapping between two time series sequences where the shortest global route is determined based on the computation and comparisons of several options at each step. While this contributes to the algorithm's high performance, the extensive computations make DTW a very time consuming algorithm. Furthermore, although DTW achieves global minimal warping path, the path is never readjusted, meaning local detail structures could be overlooked. In practice, it can often be observed that the warping path does not match the intuitive mapping, and once a point is mapped to an incorrect point, the optimal match would be permanently missed.

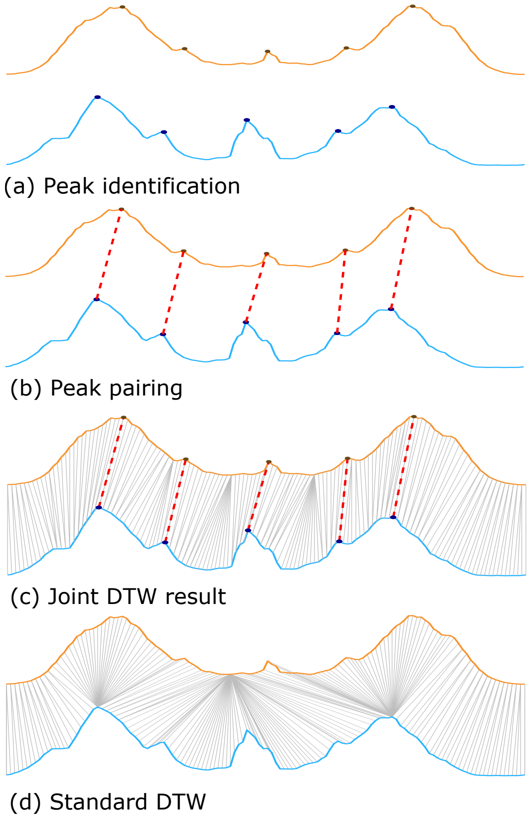
(a) Peak identification
(b) Peak pairing
(c) Joint DTW result
(d) Standard DTW

Figure 1: How time series segmentation can improve the standard DTW, (a) depicts significant feature-peak selection, (b) pairs the identified peaks, (c) shows the joint mapping results, and (d) is the mapping from standard DTW. The joint mapping results avoided some pathological warpings that occurred in standard DTW and is also more intuitive.

When humans are presented with two sequences, often times we are able to ignore temporal misalignment, evaluate differences, and zoom in on the highly similar segments to determine similarity. We can notice the key similarities in different sequences due to the significant features, such as a peak, a valley, or a characteristic slope. When we compare sequences within our visual scope, we do so on a global scale. We do not compare each value on the sequences, therefore, the process is highly efficient. This is our inspiration for improving the standard DTW. By segmenting the time series according to the peaks, each segment's warping path is more curated to the broad framework of global similarity.

The effects of segmentation are shown in Fig. 1. Shown in Fig. 1(a), we start by identifying the significant features, in this case the peaks of each time series sequence; then we match the peaks, as is shown in Fig. 1(b); and carry out DTW on each sub-sequence, the result of which is shown in Fig. 1(c). Due to the segmentation within each time series sequence, the mapping in Fig. 1(c) is more intuitive than the standard DTW in Fig. 1(d). Theoretically, any feature extracted from the data that could meaningfully segment time series can be used. Here we use peaks because it is easy to recognize and utilize, valleys would work the same way. We investigate the effect of different peak identification parameters.

The rest of this paper is organized as follows: Section II gives the background on DTW improvement techniques; Section III discuss how peak identification is done on time series; Section IV shows the effect of different peak identification parameters and how segmentation can improve the standard DTW; finally, Section V concludes this paper.

II. Background

A. Dynamic Time Warping

Dynamic Time Warping (DTW) is an algorithm for measuring the similarity between two temporal sequences which may vary in time or speed. This allows computers to find an optimal match between two given sequences. Unlike lock-step distance measures, DTW allows one-to-many mappings; this adds flexibility to the distance measurements.

Originally, DTW was used in speech recognition [4], later it was adapted to various real-world data mining problems. Given two time series sequences $Q = q_1, q_2, ..., q_i, ..., q_n$, and $C = c_1, c_2, ..., c_j, ..., c_m$, Equations 1 and 2 show the computation for Euclidean and DTW distances respectively, with Euclidean only valid for equal length sequences ($n = m$) and DTW valid for both equal and unequal length sequences.

$$dist(Euclidean) = \sqrt{\sum_{i=1}^{n}(q_i - c_i)^2} \tag{1}$$

$$D(i,j) = dist(q_i, c_j) + min \begin{cases} D(i, j-1) \\ D(i-1, j-1) \\ D(i-1, j) \end{cases} \tag{2}$$

When calculating the DTW distance, an n-by-m distance matrix is first constructed containing the distance information between all the elements from the two sequences. The distance between mapped data points q_i and c_j for DTW is computed as the Euclidean distance between them. The warping path is denoted as $W = w_1, w_2, ..., w_k, ..., w_K$. While there are exponentially many warping paths, only the minimized path is of interest [5].

The basic rules of DTW include the boundary condition, monotonicity, and continuity [6]. The boundary condition means that every element has to have a mapping component, and the first and last components from two compared sequences are always mapped. Monotonicity refers to the single direction of time, and a warping path cannot go back in time. The continuity constraint is also known as the step pattern constraint; it is where the warping path can only follow the steps allowed and cannot make any jumps.

B. Existing DTW Improvement Methods

DTW avoids the naïve injective mapping, to provide a more natural alignment between time series. However, despite its general success, the algorithm often attempt to explain variability in the y-axis of the similarity matrix by warping on the x-axis. This undesirable phenomenon is called "singularity", and could lead to pathological warping [5]. To avoid pathological warpings and to speed up the alignment procedure, many approaches have been proposed.

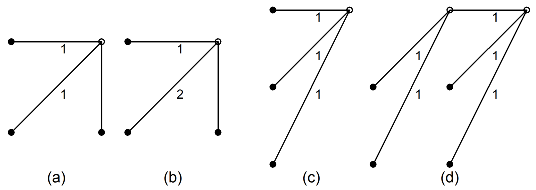

Figure 2: Step Patterns: (a) symmetric1, (b) symmetric2, (c) asymmetric, and (d) rabinerJuangStepPattern

Windowing has been used by different researchers for a long time, and was formally summarized by Berndt and Clifford [7]. It effectively prunes the corners of the matrix so that any potential warping path is bounded within a fixed margin. This method can mitigate the singularity problem to some extent, but cannot prevent it [8]. Well known global windowing constraints include *Sakoe-Chiba Band* [4], which is a slanted diagonal window, and *Itakura parallelogram* [9].

Slope weighting encourages the warping path to remain close to the diagonal. Depending on the specific weighting factor, it reduces the frequency of singularities [10].

Step pattern is another approach which encourage changes to the warping path to avoid pathological paths. Four of the well known patterns are shown in Fig. 2 [11]. Based on symmetry and slope bounds, Sakoe and Chiba proposed *symmetric1*, *symmetric2* [4], and *asymmetric* [12] approaches. The basic step pattern *symmetric1* is shown in Fig 2(a). Fig 2(b) shows the *symmetric2* step pattern, which favors the diagonal warping path similar to slope weighting. The *asymmetric* step pattern in Fig 2(c) limits time expansion to a factor of two.

Rabiner and Juang introduced *rabinerJuangStepPattern* [13] shown in Fig 2(d), which is based on the continuity constraint, slope weighting, and the state of being smooth or Boolean. We will be using the popular *symmetric1* step pattern for all the experiments supporting this paper.

Another avenue for improving the DTW algorithm is by defining tight and fast lower bounding functions to prune mappings that cannot provide a better match in the process of finding the warping path. The idea of finding lower bounds is to favor the execution time needed for calculating the similarity matrices on large datasets. Three well-known lower bounding measures are LB_Yi [14], LB_Kim [15], and LB_Keogh [16].

For two time series Q and C, LB_Yi's bounding function is defined based on the distance of all the points in Q which are greater (less) than the maximum (minimum) point in C, from the maximum (minimum) point in C, or the other way around. Depending on the three possible arrangements of the two time series: overlap, enclose, or disjoint. While Yi et al. give an approximation for indexing, the lower bounding function introduced in LB_Kim was the first to define an exact indexing. Kim et al. extracted four features from each time series: the first and last data points, and the minimum and maximum values. After comparing the extracted features of the two time series, the largest squared differences of the corresponding features, calculated at query time, is considered as the lower bound measure. When compared to the earlier works, LB_Keogh had an overall greater pruning power and could also give tighter bounding measures. Their lower bounding function is defined based on U and L, the two new time series generated from the reference time series Q, such that $U_i = max(q_{i-r}, q_{i+r})$ and $L_i = min(q_{i-r}, q_{i+r})$. Where $j-r \leq i \leq j+r$, and r is used to define the allowed warping range. Having the bounding envelope defined by U and L, the lower bounding function is defined as "the squared sum of the distances from every part of the candidate sequence C not falling within the bounding envelope, to the nearest orthogonal edge of the bounding envelope" [16].

Among other variants of the DTW method, PDTW (piece-wise DTW) [17], DDTW (derivative DTW) [5], and shapeDTW [18] are some of the more popular methods which attempt to manipulate the input time series to improve either the warping path or the processing time. The primary achievement of PDTW is to increase the speed factor by one to two orders of magnitude on average, while maintaining the accuracy of DTW. The key idea is to instead of using raw data, a piece-wise aggregated representation of the time series is fed to the DTW method. Similarly, DDTW utilizes an approximated derivative of the time series to work on a higher level of similarity between two time series, as opposed to the actual values. Utilizing derivatives compensate for the pathological warpings which are often observed because of the difference between the values of the two time series along the y-axis. A similar approach is the shapeDTW. Zhao et al. represented each temporal point q_i of a time series Q by a shape descriptor d_i which encodes the structural information of a fixed-width neighborhood of q_i. The choice of the descriptor depends on the general structure of the time series and the users' requirements. Some of the widely used descriptors are namely the slope, piece-wise aggregate approximation (PAA), discrete wavelet transform (DWT), and the histogram of oriented gradient for 1D time series (HOG1D).

The main goal of windowing, slope weighting, and step pattern is to avoid pathological warpings by trying to encourage the warping path to stay close to the diagonal rather than to stray vertical or horizontal. These global constraints, along with lower bounding can also speed up the computation process by eliminating the need for some of the calculations. Other methods utilize approximated values or dimensionality reduction to speed up computation.

III. Methodology

In the aforementioned DTW improvement methods, the rules imposed on DTW all have an equal global impact across the time series sequences, this could mean certain details are overlooked. With the goal of improving the DTW algorithm performance, we propose segmenting time series based on significant features to reduce the occurrence of pathological warping and to improve DTW performance. While the feature peaks are found globally, the effects are more local; they do not have an overall equal impact on the entire sequence. Meaning that the warping path has more freedom to grow compared to methods such as warping windows. Through detecting time series data peaks and segmenting accordingly, we add a layer of approximation before DTW distance computation. While this idea is straightforward, it is effective in avoiding pathological warpings and providing scalability.

A. Time Series Segmentation

The global optimal solution of DTW often overlooks local features in time series. For example, in Fig. 3, the peaks marked in boxes on time series Q are intuitively a match with the peaks below in time series C. When using the original DTW algorithm, peak c and c' can be directly mapped. However, a' is not only mapped to a, but also various points around a. The same is true for b' and b. As shown in Fig. 4, by identifying the peak features, we can segment each time series into four segments. As a result of the boundary condition, during computation a' and b' are specifically mapped to a and b respectively, thus minimizing mismatches and avoiding pathological warping paths. When we pair the respective peaks from both sequences, the optimal DTW mapping for each subsequence can be found. The DTW distance between two sequences is the DTW distance sum of each segmented sequence pair. The advantage of this method is that depending on the identified peaks, different segments have different tightness of constraints, so it is more flexible and adaptive with different time series.

Algorithm 1 is a simple yet flexible peak detection heuristic. The naïve definition of a peak, as this method utilizes, can be formulated as follows. The temporal index i corresponds to the peak c_i, if $c_i > c_{i-1}$ and $c_i > c_{i+1}$. In this text, we refer to this

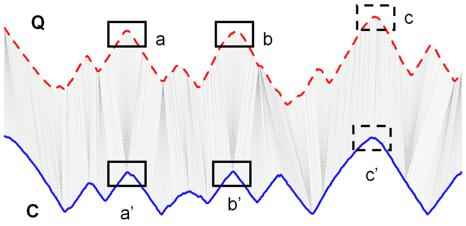

Figure 3: Unintuitive DTW mappings

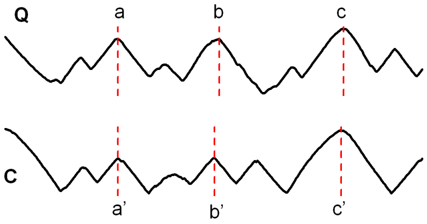

Figure 4: Segmented time series with identified peaks

definition as *candidate peaks*. This simple definition equipped with the following parameters form a peak detection method that provides the criteria necessary to distinguish *significant peaks*, which are used in segmenting time series:

- t: the threshold on the frequency domain. Any candidate peaks below t will be ignored.
- d: the minimum peak radius distance. For any peak that has already met the criteria, any adjacent peaks within a radius of d will be considered as either noise or insignificant and therefore ignored.
- n: the maximum number of peaks to be taken into account. Since the peak values will be sorted before being analyzed, this parameter can push an additional constraint on the number of peaks to be detected. That is, only the top n peaks that have met the other criteria will be selected.

Our peak detection heuristic works as follows. Initially, all candidate peaks are found and sorted based on their values (Algorithm 1 lines 3-7). Then, the threshold t on the frequency domain is applied, and all the candidate peaks below the threshold will be removed from the list (Algorithm 1 lines 8-13). For each of the remaining peaks, their neighboring peaks, within the radius of d temporal indices, will then be removed as well (Algorithm 1 lines 15-21). Note that the algorithm processes the peaks in a top-down fashion. This is to guarantee that presence of a smaller peak never justifies the removal of a larger peak. Finally, among the peaks left in the list, only the top n peaks will survive. Peaks meeting the set of constraints: t, d, and n are the selected peaks for the following segmentation.

In the worst-case, the time complexity of this algorithm, if implemented naïvely, is $max\{O((p\cdot(p+1))/2), s)\}$ where p is the number of candidate peaks and s is the time bound of the utilized sorting algorithm. The worst case refers to the situation where $d=0$, $n=\infty$, and $t=min(C)$. However, by taking the order of the indices into account, in addition to the order of the values, the time complexity would only be determined by the sorting step. Hence, the worst case running time would be decreased to $O(n\cdot\log(n))$ which reflects the complexity of a sort algorithm such as Merge Sort.

Algorithm 1 Time Series Peak Selection

Input: $c=\{c_1,\cdots,c_m\}$ time series data,
d: the minimum radius from a selected peak within which all other peaks are ignored,
t: the minimum threshold on the frequency domain below which all peaks are ignored,
n: the maximum number of peaks to be detected.
Output:
the list of peaks with both their indices and values.

```
1:  procedure FIND PEAKS
2:      $significant.peaks$, $candidates \leftarrow$ list()
3:      for all $c_i \in C$ do
4:          if $((c_i > c_{i+1})$ & $(c_i > c_{i-1}))$ then
5:              $candidates$.add(($i$, $c_i$))
6:          end if
7:      end for
8:      $peaks \leftarrow$ sortByValue($candidates$)
9:      for all $(i, c_i) \in peaks$ do
10:         if $c_i < t$ then
11:             $peaks$.remove(($i, c_i$))
12:         end if
13:     end for
14:     $indices \leftarrow peaks$.getIndices()
15:     for all $i \in indices$ do
16:         for all $j \in \{i-d,\cdots,i+d\}\setminus i$ do
17:             if $peaks$.hasIndex($j$) then
18:                 $peaks$.remove(($j$, $c_j$))
19:             end if
20:         end for
21:     end for
22:     $significant.peaks \leftarrow peaks$.getNFirstElements($n$)
23:     return $significant.peaks$;
24: end procedure
```

IV. Experiments

In this section, we explore how different peak identification parameters in time series segmentation can effect Dynamic Time Warping. The experiments will be done using datasets from the UCR repository [19]. The threshold for peak identification t is given values of the first quartile (Q1), median (M) and the third quartile (Q3). The minimum peak identification radius d is given values: $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, $\frac{1}{16}$, $\frac{1}{32}$, $\frac{1}{64}$, and $\frac{1}{128}$ of the time series length l. For the sake of simplicity, d values in figures will only be referred to with the denominator.

When the peak selection radius d is large or when the peak threshold t is large, fewer peaks are identified, which means fewer segments. When no peaks are detected, and no segment is found, it simply becomes standard DTW. In contrast, when the peak detection radius is small or when the

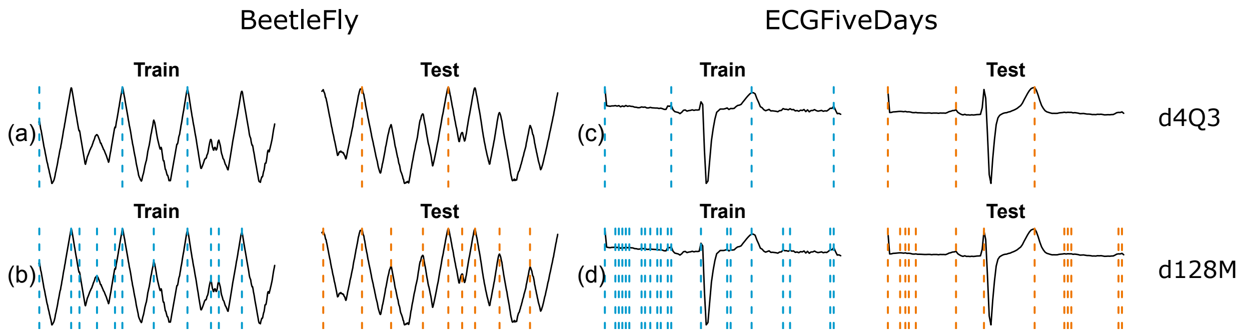

Figure 5: (a) and (b) are the same train and test sample with the same label from dataset "BeetleFly". (c) and (d) are the same train and test sample with the same label from dataset "ECGFiveDays". (a) and (c) has peak identification parameters $d{=}\frac{1}{4}l$ with $t = Q3$, (b) and (d) has peak identification parameters $d{=}\frac{1}{128}l$ with $t = M$.

peak threshold is low, there are more peaks, which leads to more segments and potentially better avoidance of pathological warpings. However, more segments in time series sequences introduce risks of identifying false peaks, which could lead to bad segmentation and worsen DTW performance.

Fig. 5 shows two sets of train and test time series from datasets "BeetleFly" and "ECGFiveDays". The identified peaks are labeled with dashed lines. Here we demonstrate the effect of peaks selection parameters on different datasets. Apparently, "BeetleFly" sequences are densely-peaked, whereas "ECGFiveDays" is very sparsely-peaked. In our experiments, the best peak selection parameter for dataset "BeetleFly" is $d{=}\frac{1}{128}l$, and $t = M$, and the best peak selection parameter for dataset "ECGFiveDays" is $d{=}\frac{1}{4}l$, and $t = Q3$. For densely-peaked datasets, if the parameters are set to find too few peaks, the performance is greatly influenced by the actual value of each peak. Which could potentially introduce uncertainty and errors. On the other hand, when the parameters are set to allow too many peaks, sparsely-peaked sequences would identify too many false peaks. Also leading to the deterioration in performance.

The main goal of segmenting time series as a means to improve the standard DTW is to avoid pathological warpings. Fig. 6 uses the same example as Fig. 5(c), with step pattern "symmetric1". Fig. 6(a) shows the standard DTW mapping, with the DTW distance of 23.68683. Fig. 6(b) shows the joint result of three segments of time series DTW. The joint warping path avoided all the pathological warpings, and the DTW distance of each segment are 0.384774, 0.159536, and 0 respectively.

Fig. 7 shows the processing time with different peak identification parameters on datasets: "ArrowHead", "BeetleFly", "Car", "Coffee", "ECGFiveDays", "Ham", "Herring", "Lighting7", "Meat", "OliveOil", "Plane", "ShapeletSim", "ToeSegmentation1", "Trace", "Wine". Initially, the more potential peaks there are the shorter the processing time. However, once the peak identification radius value d gets past the value $\frac{1}{32}l$, the overhead of peak identification becomes more significant. Given the same d, the peak threshold value $Q3$ has overall more efficient performance, especially for larger d values.

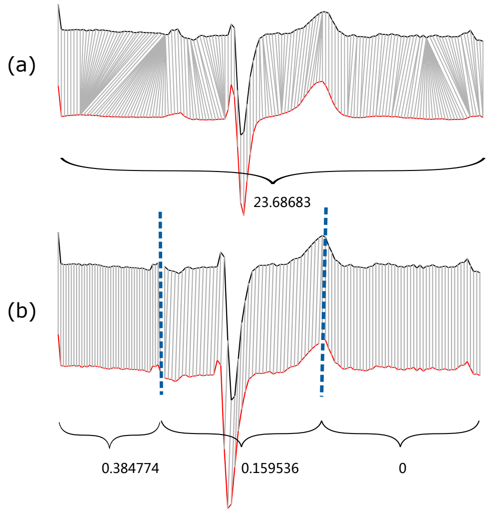

Figure 6: Standard DTW compared against joint DTW from time series segmentation

V. Conclusion

In this paper, we proposed the idea of improving the current Dynamic Time Warping algorithm with time series segmentation and also explored the effect of different peak selection parameters. Given the wide variety of time series datasets, it is near impossible to obtain a set of peak identification parameters applicable to all datasets. However, with appropriate segmentation, most pathological warping can be avoided without imposing an overpowering global constraint. An important aspect is to gain prior knowledge of the data on hand. Identifying too few peaks on a densely-peaked dataset, or too many peaks on a sparsely-peaked dataset would lead to the deterioration of performance.

Our next step is to predetermine suitable parameters for peak identification for different datasets, and also a better way to pair the identified peaks from two time series sequences. This can help us avoid pathological warping paths, and also process each time series segment simultaneously, which would lead to an improved version of Dynamic Time Warping.

ACKNOWLEDGMENT

This project has been supported in part by funding from the Division of Advanced Cyber infrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Astronomical Sciences within the Directorate for Mathematical and Physical Sciences, and the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF award #1443061. It was also supported in part by funding from the Heliophysics Living With a Star Science Program, under NASA award #NNX15AF39G.

The SDO data is courtesy of NASA/SDO (https://sdo.gsfc.nasa.gov/) and the Atmospheric Imaging Assembly (AIA), Extreme Ultraviolet Variability Experiment (EVE), and Helioseismic and Magnetic Imager (HMI) science teams.

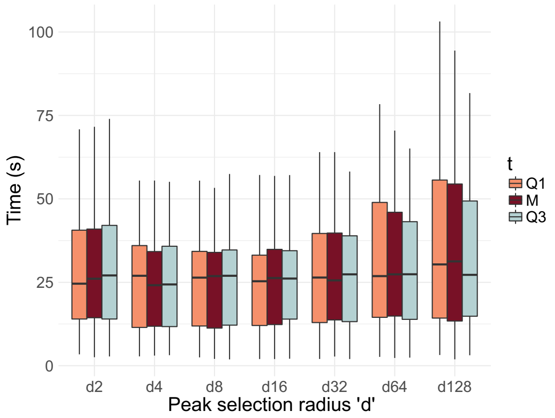

Figure 7: Effect of peak identification parameters on time series segmentation efficiency.

REFERENCES

[1] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, “Experimental comparison of representation methods and distance measures for time series data,” *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[2] P. Ranacher and K. Tzavella, “How to compare movement? a review of physical movement similarity measures in geographic information science and beyond,” *Cartography and geographic information science*, vol. 41, no. 3, pp. 286–307, 2014.

[3] J. Serra and J. L. Arcos, “An empirical evaluation of similarity measures for time series classification,” *Knowledge-Based Systems*, vol. 67, pp. 305–314, 2014.

[4] H. Sakoe and S. Chiba, “Dynamic programming algorithm optimization for spoken word recognition,” *IEEE transactions on acoustics, speech, and signal processing*, vol. 26, no. 1, pp. 43–49, 1978.

[5] E. J. Keogh and M. J. Pazzani, “Derivative dynamic time warping,” in *Proceedings of the 2001 SIAM International Conference on Data Mining*, pp. 1–11, SIAM, 2001.

[6] M. Müller, *Information retrieval for music and motion*, vol. 2. Springer, 2007.

[7] D. J. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series.,” in *KDD workshop*, vol. 10, pp. 359–370, Seattle, WA, 1994.

[8] M. Biba and F. Xhafa, *Learning Structure and Schemas from Documents*, vol. 375. Springer, 2011.

[9] F. Itakura, “Minimum prediction residual principle applied to speech recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, no. 1, pp. 67–72, 1975.

[10] J. Kruskall and M. Liberman, “The symmetric time warping algorithm: From continuous to discrete. time warps, string edits and macromolecules,” 1983.

[11] T. Giorgino *et al.*, “Computing and visualizing dynamic time warping alignments in r: the dtw package,” *Journal of statistical Software*, vol. 31, no. 7, pp. 1–24, 2009.

[12] H. Sakoe and S. Chiba, “Comparative study of dp-pattern matching techniques for speech recognition,” in *1973 Tech. Group Meeting Speech, Acoust. Soc. Japan*, 1973.

[13] L. R. Rabiner and B.-H. Juang, “Fundamentals of speech recognition,” 1993.

[14] B.-K. Yi, H. Jagadish, and C. Faloutsos, “Efficient retrieval of similar time sequences under time warping,” in *Data Engineering, 1998. Proceedings., 14th International Conference on*, pp. 201–208, IEEE, 1998.

[15] S.-W. Kim, S. Park, and W. W. Chu, “An index-based approach for similarity search supporting time warping in large sequence databases,” in *Data Engineering, 2001. Proceedings. 17th International Conference on*, pp. 607–614, IEEE, 2001.

[16] E. Keogh and C. A. Ratanamahatana, “Exact indexing of dynamic time warping,” *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[17] E. J. Keogh and M. J. Pazzani, “Scaling up dynamic time warping for datamining applications,” in *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 285–289, ACM, 2000.

[18] J. Zhao and L. Itti, “shapedtw: shape dynamic time warping,” *arXiv preprint arXiv:1606.01601*, 2016.

[19] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista, “The ucr time series classification archive,” July 2015. www.cs.ucr.edu/~eamonn/time_series_data/.