Windowing optimization approach for Piecewise Dynamic Time Warping in time series classification

Vanel Steve SIYOU FOTSO[12], Engelbert MEPHU NGUIFO[12], Philippe VASLIN[12]

[1] Clermont University, Blaise Pascal University, LIMOS, BP 10448, F-63000 CLERMONT-FERRAND

[2] CNRS, UMR 6158, LIMOS, F-63173 AUBIERE

{siyou, mephu, vaslin} @ isima.fr

Keywords : *classification, time series, heuristic.*

1 Introduction

An important task while analyzing time series, is their comparison that can be done along two main ways. Either the comparison method considers that there is no time distortion : it is the case of Euclidian distance (ED), or it considers that some small time distortions exist between time axes of time series. It is the case of the Dynamic Time Warping alignment algorithm (DTW). As time distortion often exists between time series, DTW often yields better results than ED [1]. However, DTW have two major drawbacks : the comparison of two time series is time-consuming [3] and DTW sometimes produces pathological alignments [2]. Piecewise DTW (PDTW) was introduced with the aim to speed up the computation of DTW, which depends on the length of the time series. PDTW proposes to split a time series into consecutive fixed-length segments and to compute the mean of each segment. Then, mean values are used instead of datapoints to compare the time series. In practice, the exhaustive approach is the brute-force (BF) approach that explores all the possible values of this parameter. However, this is time consuming especially with long time series data. So, the question is how to automatically fix this parameter without a considerable decrease of classification accuracy ? In this paper, we propose a parameter free heuristic (FDTW) to align piecewise aggregate time series with Dynamic Time Warping that approximates the optimal value of the segments number to be considered during the alignment.

2 Parameter free heuristic

The idea of our heuristic is the following :

1. We choose N_c candidates distributed in the space of all possible values to ensure that we are going to have small, medium and large values as candidates. The candidates values are : n, $n - \left\lfloor \frac{n}{N_c} \right\rfloor$, $n - 2 \times \left\lfloor \frac{n}{N_c} \right\rfloor$, ..., $n - N_c \times \left\lfloor \frac{n}{N_c} \right\rfloor$. For instance, if the length of time series is $n = 12$ and the number of candidates is $N_c = 4$, we are going to select the candidates 12, 9, 6, 3.

$$1, 2, [3], 4, 5, [6], 7, 8, [9], 10, 11, [12]$$

2. We evaluate the classification error with $1NNPDTW$ for each candidate that we have previously chosen and we select the candidate that has the minimal classification error : it is the best candidate. In our example, we may suppose that we get the minimal value with the candidate 6 it is thus the best candidate at this step.

$$1, 2, 3, 4, 5, [6], 7, 8, 9, 10, 11, 12$$

3. We respectively look between the predecessor (i.e., 3 here) and successor (i.e., 9 here) of the best candidate for a number of segments with a lower classification error. This number of segments corresponds to a local minimum. In our example, we are going to test the values 4, 5, 7 and 8 to see if there is a local minimum.

4. We restart at step one, while choosing differents candidates during each iteration to ensure that we return a good local minimum. We fix the number of iterations to $\lfloor log(n) \rfloor$. At each iteration the first candidate is $n - (number\_of\_iteration - 1)$.

To resume, in the worst case, we test the N_c first candidates to find the best one. Then, we test $\frac{2n}{N_c}$ other candidates to find the local minimum. We finally perform $nb(N_c) = N_c + \frac{2n}{N_c}$ tests. The number of tests that we have to perform is a function of the number of candidates. How many candidates should we consider to reduce the number of tests ? The first derivative of the function nb vanishes when $N_c = \sqrt{2n}$ and the second derivative is positive so the minimal number of tests is done when the number of candidates $N_c = \sqrt{2n}$.

3 Experiments and discussion

3.1 Datasets

The performance of our heuristic FDTW has been tested on 34 datasets of the UCR time series datamining archive [1], which provides a large collection of datasets that cover various categories of domains. The 34 datasets possess between 2 and 50 classes, the length of the time series varies from 24 to 1882, the training sets contain between 20 and 1000 time series and the testing sets contain between 28 and 6174 time series.

3.2 Discussions

The experiments showed that FDTW is faster than BF algorithm. Moreover, they show the closeness between BF and FDTW classification error and that FDTW allows to find the minimum error for 8 data sets (Coffee, ECGFiveDays, Gun-point, ItalyPowerDemand, OliveOil, Synthetic control, Trace, Two patterns).

4 Conclusion

FDTW allows to reduce the storage space and the processing time of time series classification without decreasing the alignment quality. As a perspective, we plan to use piecewise aggregate to find the number of segments to be considered for symbolic representations of time series like SAX, ESAX, SAX-TD.

Références

[1] Yanping Chen, Eamonn Keogh, Bing Hu, Nurjahan Begum, Anthony Bagnall, Abdullah Mueen, and Gustavo Batista. The ucr time series classification archive, July 2015. www.cs.ucr.edu/ eamonn/time_series_data/.

[2] E J Keogh and M J Pazzani. Derivative dynamic time warping. *Proceedings of the 1st SIAM International Conference on Data Mining*, pages 1–11, 2001.

[3] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. *Proceedings of the 18th ACM SIGKDD*, pages 262–270, 2012.