Jain, Brijnesh; Froese, Vincent; Schultz, David

An average-compress algorithm for the sample mean problem under dynamic time warping

Journal of Global Optimization

Provided in Cooperation with:
Springer Nature

Suggested Citation: Jain, Brijnesh; Froese, Vincent; Schultz, David (2023) : An average-compress algorithm for the sample mean problem under dynamic time warping, Journal of Global Optimization, ISSN 1573-2916, Springer US, New York, NY, Vol. 86, Iss. 4, pp. 885-903, https://doi.org/10.1007/s10898-023-01294-9

Journal of Global Optimization (2023) 86:885–903
https://doi.org/10.1007/s10898-023-01294-9

An average-compress algorithm for the sample mean problem under dynamic time warping

Brijnesh Jain[1] · **Vincent Froese**[2] · **David Schultz**[3]

Received: 13 October 2020 / Accepted: 5 May 2023 / Published online: 3 June 2023

Abstract
Computing a sample mean of time series under dynamic time warping is NP-hard. Consequently, there is an ongoing research effort to devise efficient heuristics. The majority of heuristics have been developed for the constrained sample mean problem that assumes a solution of predefined length. In contrast, research on the unconstrained sample mean problem is underdeveloped. In this article, we propose a generic average-compress (AC) algorithm to address the unconstrained problem. The algorithm alternates between averaging (A-step) and compression (C-step). The A-step takes an initial guess as input and returns an approximation of a sample mean. Then the C-step reduces the length of the approximate solution. The compressed approximation serves as initial guess of the A-step in the next iteration. The purpose of the C-step is to direct the algorithm to more promising solutions of shorter length. The proposed algorithm is generic in the sense that any averaging and any compression method can be used. Experimental results show that the AC algorithm substantially outperforms current state-of-the-art algorithms for time series averaging.

Keywords Time series averaging · Fréchet function · Heuristic · Nonconvex optimization

1 Introduction

Time series such as stock prices, climate data, energy usages, sales, biomedical measurements, and biometric data are sequences of time-dependent observations that often vary in

B. Jain was funded by the DFG project JA 2109/4-2.

✉ Vincent Froese
vincent.froese@tu-berlin.de

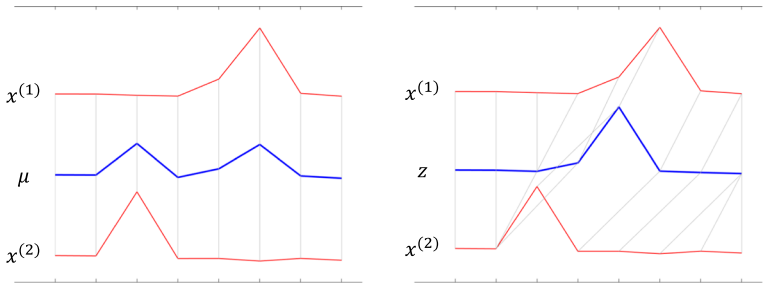
Brijnesh Jain
brijnesh.jain@oth-regensburg.de

David Schultz
david.schultz@dai-labor.de

[1] Department of Computer Science and Mathematics, OTH Regensburg, Regensburg, Germany

[2] Algorithmics and Computational Complexity, Faculty IV, TU Berlin, Berlin, Germany

[3] Distributed Artificial Intelligence Laboratory, Faculty IV, TU Berlin, Berlin, Germany



(a) Arithmetic mean (b) DTW Mean

Fig. 1 Mean time series (blue) of the two sample time series $x^{(1)}$ and $x^{(2)}$ shown in red. Both time series have a single peak but are out of phase and slightly vary in speed. We may think of $x^{(1)}$ and $x^{(2)}$ as the daily average temperature of some region during the summer at two different years. Based on this information, a typical summer of this region has a single extreme heat wave. The arithmetic mean $\mu = (x^{(1)} + x^{(2)})/2$ in Fig. 1a has two attenuated peaks suggesting that a typical summer has two moderate heat waves. In contrast, the DTW mean z in Fig. 1b captures the characteristic properties of $x^{(1)}$ and $x^{(2)}$ and shows a single peak as a representative summary of both sample peaks. (Color figure online)

temporal dynamics, that is in length, speed, and shifts in phase. For example, the same word can be uttered with different speaking speeds. Similarly, monthly temperature or precipitation extremes of certain regions can differ in duration and may occur out of phase for a period of a few weeks.

To account for temporal variations in proximity-based time series mining, the *dynamic time warping* (DTW) distance is often the preferred choice of proximity measure [1, 3, 4]. An intricate problem in DTW-based time series mining is time series averaging. The problem consists in finding a typical representative that summarizes a sample of time series. Different forms of time series averaging have been applied to improve nearest neighbor classifiers [20, 31, 32], to accelerate similarity search [37], and to formulate k-means clustering in DTW spaces [17, 30, 32, 36]. Figure 1 presents an example illustrating why the arithmetic mean can be inappropriate for time series averaging and motivates a concept of mean under dynamic time warping that can cope with temporal variations.

Time series averaging itself and as a subroutine of data mining tasks is inspired by the fundamental concept of mean in statistical inference. One central path in statistical inference departs from the mean, then leads via the normal distribution and the Central Limit Theorem to statistical estimation using the maximum likelihood method. The maximum likelihood method in turn is a fundamental approach that provides probabilistic interpretations to many pattern recognition methods.

This central path is well-defined in Euclidean spaces, but becomes obscure in mathematically less structured distance spaces. Since an increasing amount of non-Euclidean data is being collected and analyzed in ways that have not been realized before, statistics is undergoing an evolution [26]. Examples of this evolution are contributions to statistical analysis of shapes [5, 13, 24], complex objects [29], tree-structured data [14, 40], graphs [15, 19, 23], and time series [7, 17, 32].

Though the volume of time series data currently collected exceeds those of the other data structures mentioned above, the concept of a mean in DTW spaces is least understood. However, a better understanding of time series averaging is the first step towards devising sound pattern recognition methods based on time series averaging such as k-means clustering.

Examples of how the lack of a clear understanding of time series averaging may lead the field astray can be found in [7, 21].

As for other non-Euclidean distance spaces, the standard approach to time series averaging in DTW spaces is based on an idea by Fréchet [16]: Suppose that $\mathcal{S} = \{x_1, \ldots, x_N\}$ is a sample of N time series. A sample mean of $\mathcal{S}$ is any time series μ that globally minimizes the Fréchet function

$$F : \mathcal{U} \to \mathbb{R}, \quad z \mapsto \frac{1}{N} \sum_{i=1}^{N} \mathrm{dtw}(z, x_i)^2,$$

where $\mathrm{dtw}(x, y)$ is the DTW-distance and $\mathcal{U}$ is a set of time series of finite length. The search space $\mathcal{U}$ typically takes two forms:

1. Unconstrained form: $\mathcal{U}$ is the set of all time series of finite length.
2. Constrained form: $\mathcal{U}$ is the set of all time series of length n.

A sample mean is guaranteed to exist in either case but may not be unique [22]. In addition, computing a sample mean is NP-hard [8].

In general, data mining algorithms such as k-means, vector quantization, and anomaly detection that rely on a sample mean typically expect a *true* sample mean in their objective function, that is a sample mean that minimizes the variance in a Euclidean sense or more generally the Fréchet function in any distance space. As experiments indicate [7], better sample means result in better solutions of the k-means algorithm. The k-means algorithm needs to recompute the sample mean several times. The error of suboptimal solutions propagates during evolution of the k-means algorithm resulting in possibly distorted clusters. Similarly, a suboptimal sample mean in anomaly detection could result in lower precision and recall.

Consequently, there is an ongoing research on devising heuristics for minimizing the Fréchet function. Most contributions focus on devising and applying heuristics for the constrained sample mean problem. State-of-the-art algorithms are stochastic subgradient methods [35], majorize-minimize algorithms [17, 30, 35], and soft-DTW [11]. In contrast, only few work has been done for solving the unconstrained sample mean problem. One algorithm is an (essentially optimal) dynamic program that finds global solutions of the unconstrained problem in exponential time [7]. A second algorithm is a heuristic, called adaptive DBA (ADBA) [28]. This algorithm uses a majorize-minimize algorithm (DBA) as a base-algorithm and iteratively refines subsequences to improve the solution quality.

The unconstrained problem is more challenging than the constrained one in the following sense: The constrained problem for length $n \in \mathbb{N}$ can be modelled as an optimization problem in a Euclidean space $\mathbb{R}^n$, where we have efficient local optimization techniques such as subgradient methods.[1]

In contrast, the unconstrained problem can be modelled as a finite collection of constrained problems for sample means of length $n = 1, \ldots, n_0$ where the maximum possible length n_0 of a sample mean is bounded by the Reduction Theorem [22]. The maximum length n_0 depends linearly on the lengths of the sample time series. A naive approach to address the unconstrained problem is to use a mean algorithm for the constrained problem and run it for all lengths $n = 1, \ldots, n_0$ and return the solution with lowest Fréchet function value.

Apart from the fact that the naive approach is likely to be computationally intractable, there is the additional problem of how to choose an initial mean time series of desired length for each constrained mean algorithm. Current heuristics choose as initial point a random time series from the sample or the medoid of the sample. The initial length is therefore restricted to

[1] The majorize-minimize algorithms [17, 30] such as DBA are subgradient methods [35].

the lengths of the sample time series. One approach to obtain initial points of arbitrary length is generating them from a normal distribution. However, this method performed considerably worse in experiments [30].

In this work, we propose a generic average-compress (AC) algorithm for the unconstrained sample mean problem that addresses these issues. The AC algorithm repeatedly alternates between an averaging (A-step) and a compression (C-step). The A-step requires a time series as initial guess, minimizes the Fréchet function, and returns an approximate solution as output. The C-step compresses the approximation of the A-step to obtain an improved solution. The compressed solution of the C-step serves as initial guess of the A-step in the next iteration.

To ensure that the C-step returns an improved solution, a whole compression chain is efficiently computed and the best compression is selected. The AC algorithm can be initialized with a time series from the sample and iteratively seeks for promising solutions of shorter length using a compression technique. Thus, compression addresses both issues, finding a suitable length of the approximate solution for the unconstrained sample mean problem and finding suitable initial points for current state-of-the-art sample mean algorithms. In principle, any averaging algorithm and any compression method can be applied. Here, we propose a compression method that minimizes the squared DTW error between original and compressed time series. Empirical results suggest that the AC scheme substantially outperforms state-of-the-art heuristics including ADBA.
Our main contributions can be summarized as follows.

1. We propose a generic average-compress (AC) algorithm for the unconstrained sample mean problem under Dynamic Time Warping,
2. we empirically evaluate different configurations of the AC algorithm on 85 benchmark data sets from the UCR archive [10], using two compression techniques and multiple state-of-the art averaging heuristics,
3. we empirically show that on average the AC algorithm considerably outperforms current state-of-the-art averaging algorithms on the UCR benchmark data sets.

This article is organized as follows: Sect. 2 describes the AC algorithm. In Sect. 3 we present and discuss empirical results. Finally, Sect. 4 concludes with a summary of the main findings and an outlook for future research.

2 Average-compress algorithm

In this section, we develop an average-compress (AC) algorithm for approximately solving the unconstrained sample mean problem. To this end, we first introduce the DTW-distance (Sect. 2.1), the concept of a sample mean under DTW (Sect. 2.2), and compressions (Sect. 2.3). Thereafter, we describe the AC algorithm in Sect. 2.4.

2.1 Dynamic time warping

For a given $n \in \mathbb{N}$, we write $[n] = \{1, \ldots, n\}$. A *time series* is a sequence $x = (x_1, \ldots, x_n)$ with elements $x_i \in \mathbb{R}$ for all $i \in [n]$. We denote the *length* of time series x by $|x| = n$, the set of time series of length n by $\mathcal{T}_n$, and the set of all time series of finite length by $\mathcal{T}$. Consider the $(m \times n)$-grid defined as

$$[m] \times [n] = \{(i, j) \,:\, i \in [m],\, j \in [n]\}\,.$$

A *warping path* of order $m \times n$ and length ℓ is a sequence $p = (p_1, \ldots, p_\ell)$ through the grid $[m] \times [n]$ consisting of ℓ points $p_l = (i_l, j_l) \in [m] \times [n]$ such that

1. $p_1 = (1, 1)$ and $p_\ell = (m, n)$
2. $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ for all $l \in [\ell - 1]$.

The first condition is the boundary condition and the second condition is the step condition of the DTW-distance. We denote the set of all warping paths of order $m \times n$ by $\mathcal{P}_{m,n}$. Suppose that $p = (p_1, \ldots, p_\ell) \in \mathcal{P}_{m,n}$ is a warping path with points $p_l = (i_l, j_l)$ for all $l \in [\ell]$. Then p defines an *expansion* (or warping) of the time series $x = (x_1, \ldots, x_m)$ and $y = (y_1, \ldots, y_n)$ to the length-ℓ time series $\phi_p(x) = (x_{i_1}, \ldots, x_{i_\ell})$ and $\psi_p(y) = (y_{j_1}, \ldots, y_{j_\ell})$. By definition, the length ℓ of a warping path satisfies $\max(m, n) \leq \ell \leq m + n$.

The *cost* of warping time series x and y along warping path p is defined by

$$C_p(x, y) = \left\| \phi_p(x) - \psi_p(y) \right\|^2 = \sum_{(i,j) \in p} \left(x_i - y_j \right)^2,$$

where $\|\cdot\|$ denotes the Euclidean norm and ϕ_p and ψ_p are the expansions defined by p. The *DTW-distance* of x and y is

$$\mathrm{dtw}(x, y) = \min \left\{ \sqrt{C_p(x, y)} \, : \, p \in \mathcal{P}_{m,n} \right\}.$$

A warping path p with $C_p(x, y) = \mathrm{dtw}^2(x, y)$ is called an *optimal warping path* of x and y. By definition, the DTW-distance minimizes the Euclidean distance between all possible expansions that can be derived from warping paths. Computing the DTW-distance and deriving an optimal warping path is usually solved via dynamic programming [34, 39].

2.2 Sample means under DTW

Let $\mathcal{S} = \{x_1, \ldots, x_N\}$ be a sample of N time series $x_i \in \mathcal{T}$. Note that $\mathcal{S}$ is a multiset that allows multiple instances of the same elements. A sample mean of $\mathcal{S}$ is any time series that minimizes the Fréchet function [16]

$$F : \mathcal{U} \to \mathbb{R}, \quad z \mapsto \frac{1}{N} \sum_{i=1}^{N} \mathrm{dtw}(z, x_i)^2,$$

where $\mathcal{U} \subseteq \mathcal{T}$ is a subset of time series. The value $F(z)$ is the *Fréchet variation* of sample $\mathcal{S}$ at z. The infimum $\inf_z F(z)$ serves as a measure of variability of $\mathcal{S}$. Here, the search space $\mathcal{U}$ takes one of the following two forms: (i) $\mathcal{U} = \mathcal{T}$ and (ii) $\mathcal{U} = \mathcal{T}_m$. We refer to (i) as the unconstrained and to (ii) as the constrained sample mean problem. Note that the constrained formulation only restricts the length of the candidate solutions, whereas there is no length restriction on the sample time series to be averaged.

A sample mean exists in either case but is not unique in general [22]. This result implies that F attains its infimum (has a unique minimum). However, computing a sample mean is NP-hard [8]. The implication is that we often need to resort to heuristics that return useful solutions within acceptable time.

We briefly describe two state-of-the-art algorithms for the constrained sample mean problem: a stochastic subgradient method (SSG) [35] and a majorize-minimize algorithm (DBA) [17, 30]. For a detailed description of both algorithms, we refer to [35].

To present the update rule of both algorithms in a compact form, we introduce the notions of warping and valence matrix as proposed by [35]. Suppose that $p \in \mathcal{P}_{m,n}$ is a warping

Algorithm 1 Stochastic Subgradient Method

procedure SSG($\eta, m, x_1, \ldots, x_N$)
  initialize solution $z \in \mathcal{T}_m$
  initialize best solution $z_* = z$
  repeat
    reshuffle order of sample time series
    for $i \leftarrow 1$ **to** N **do**
      compute optimal warping path p_i of z and x_i
      compute valence matrix V_i of p_i
      compute warping matrix W_i of p_i
      update solution z according to the rule
$$z \leftarrow z - 2\eta\,(V_i z - W_i x_i)$$
    record best solution $z_* = \operatorname{argmin}\{F(z_*), F(z)\}$
  until termination
  return z_*

path. The *warping matrix* of p is the zero-one matrix $W = (w_{ij}) \in \{0, 1\}^{m \times n}$ with elements

$$w_{ij} = \begin{cases} 1 & (i, j) \in p \\ 0 & \text{otherwise} \end{cases}.$$

The *valence matrix* of warping path p is the diagonal matrix $V = (v_{ij}) \in \mathbb{N}^{m \times m}$ with positive diagonal elements

$$v_{ii} = \sum_{j=1}^{n} w_{ij}.$$

Suppose that z and x are time series of length $|z| = m$ and $|x| = n$. Then W warps x onto the time axis of z. Each diagonal element v_{ii} of V counts how many elements of x are warped to element z_i.

Stochastic Subgradient Algorithm. Subgradient methods for time series averaging have been proposed by [35]. Algorithm 1 outlines a vanilla version of the SSG algorithm with constant learning rate η. In practice, more sophisticated stochastic subgradient variants such as Adam [27] are preferred. The input of Algorithm 1 are a learning rate η, a length-parameter m of the constrained search space, and a sample $x_1, \ldots, x_N$ of time series to be averaged. The output is a time series with lowest Fréchet variation that has been encountered during optimization.

Majorize-Minimize Algorithm. Majorize-minimize algorithms for time series averaging have been proposed in the 1970s mainly by Rabiner and his co-workers with speech recognition as the primary application [32, 42]. The early approaches fell largely into oblivion and where successively rediscovered, consolidated, and improved in a first step by Abdulla et al. [2] in 2003 and then finalized in 2008 by Hautamaki et al. [17]. In 2011, Petitjean et al. [30] reformulated, explored, and popularized the majorize-minimize algorithm by Hautamaki et al. [17] under the name DTW Barycenter Averaging (DBA).

Algorithm 2 describes the DBA algorithm. It takes a length-parameter m and a sample of time series as input and returns the candidate solution of the last iteration as output. The DBA algorithm terminates after a finite number of iterations in a local minimum of the Fréchet function [35].

Algorithm 2 DBA Algorithm

procedure DBA($m, x_1, \ldots, x_N$)
  initialize solution $z \in \mathcal{T}_m$
  repeat
    `//*** Majorize`
    for $i \leftarrow 1$ **to** N **do**
      compute optimal warping path p_i of z and x_i
      compute valence matrix V_i of p_i
      compute warping matrix W_i of p_i
    `//*** Minimize`
    update solution z according to the rule

$$z \leftarrow \left(\sum_{i=1}^{N} V_i\right)^{-1} \left(\sum_{i=1}^{N} W_i x_i\right)$$

  until termination
  return z

Algorithm 3 Adaptive Scaling

1: **procedure** ADA(x)
2: $x' \leftarrow x$ `// current compression`
3: **repeat**
4: $i \in \operatorname{argmin}\{|x'_j - x'_{j+1}| : j < |x|\}$
5: $x' \leftarrow \text{MERGE}(x', i)$
6: $\mathcal{C}(x) \leftarrow \mathcal{C}(x) \cup \{x'\}$
7: **until** $|x'| = 1$
8: **return** $\mathcal{C}(x)$
9:
10: **procedure** MERGE(x', i)
11: $z \leftarrow x'$
12: replace z_i by $(x'_i + x'_{i+1})/2$
13: delete z_{i+1}
14: **return** z

2.3 Compressions

Let $x \in \mathcal{T}$ be a time series of length n. A compression of x is a time series x' of length $m \leq n$ that maintains some desirable problem-specific properties of x. By definition, x is also a compression of itself. A *compression chain* of x is a sequence $\mathcal{C}(x) = \left(x'_1, \ldots, x'_k\right)$ of $k \in [n]$ compressions x'_i of x such that

$$1 \leq \left|x'_1\right| < \left|x'_2\right| < \cdots < \left|x'_k\right| \leq n.$$

There are numerous compression methods such as principal component analysis, discrete Fourier transform, discrete wavelet transform, and many more. Here, we consider two simple methods: adaptive scaling (ADA) and minimum squared DTW error (MSE).

Adaptive Scaling. Algorithm 3 describes ADA. The procedure takes a time series $x = (x_1, \ldots, x_n)$ as input and returns a compression chain $\mathcal{C}(x)$ consisting of n compressions of x of length 1 to n. To compress a time series x'_{k+1} of length $k+1$ to a time series x'_k of length k, ADA merges two consecutive elements with minimal distance. The merge subroutine in Algorithm 3 replaces these two consecutive time points by their average.

Finding the smallest distance in Line 4 takes $O(\left|x'\right|)$ time. In each iteration, the length of x' is reduced by one. Thus, the complexity of computing all n compressions is $O(n^2)$.

Minimum Squared DTW Error Compression. The second compression method computes a time series of a given length such that the squared DTW error is minimized. Let $x \in \mathcal{T}$ be a time series of length n and let $m < n$. We call each

$$x' \in \operatorname{argmin} \left\{ \operatorname{dtw}(x, z)^2 \,:\, z \in \mathcal{T}_m \right\}$$

an MSE compression of x of length m. Observe that the MSE compression problem for x is the constrained sample mean problem of the sample $\mathcal{S} = \{x\}$. Algorithm 4 outlines MSE compression.

It is not hard to see that for a compression x', an optimal warping path p between x and x' warps every element of x to exactly one element of x', that is, $\phi_p(x) = x$. Thus, we can write

$$\begin{aligned} p = \big(&(1,1), \ldots, (i_1, 1), (i_1 + 1, 2), \ldots, \\ &(i_1 + i_2, 2), \ldots, (n - i_m), \ldots, (n, m) \big), \end{aligned} \tag{1}$$

where $\sum_{l=1}^{m} i_l = n$. Let $d_l = \sum_{j=1}^{l} i_j$, $l \in [m]$, and $d_0 = 0$. The squared DTW error of x' is

$$\operatorname{dtw}(x, x')^2 = \sum_{l=1}^{m} \sum_{i=d_{l-1}+1}^{d_l} (x_i - x'_l)^2. \tag{2}$$

MSE compression is also known as *adaptive piecewise constant approximation* [9] and as *segmentation* problem [38]. It can be solved exactly via dynamic programming in $O(n^2 m)$ time [6]. Moreover, the dynamic program allows to find all n compressions (for each length $m = 1, \ldots, n$) in $O(n^3)$ time by running it once for $m = n$ (as it is done in Algorithm 4).

Interestingly, MSE compression is also related to one-dimensional k-means clustering. To see this relationship, consider an optimal warping path between the compression x' and the original time series x as in Eq. (1). Then, the squared DTW error is minimal for $x'_l = (x_{d_{l-1}+1} + \cdots + x_{d_l})/i_l$. Thus, finding an MSE compression x' of length k can also be seen as a one-dimensional k-means clustering problem, where every cluster consists of a *consecutive* subsequence of elements in x. Indeed, the dynamic program in Algorithm 4 is the same as for one-dimensional k-means [41] (without previously sorting the elements in x).

To reduce the computational complexity, several heuristics and approximations for MSE compression have been proposed [9, 25, 38, 43]. Also ADA compression can be regarded as a heuristic for MSE compression since it greedily averages two consecutive elements.

To conclude, with MSE compression, we consider an exact solution method to a sound compression problem and with ADA compression, we consider a fast heuristic. Among the various heuristics, we have chosen ADA compression because it has been successfully tested for improving approximate solutions of the constrained sample mean problem [30].

2.4 The average-compress algorithm

In this section, we assemble the pieces of the previous sections and propose a generic average-compress (AC) algorithm for approximately solving the unconstrained sample mean problem.

AC Algorithm. The AC algorithm alternates between averaging (A-step) and compression (C-step). For this purpose, any averaging algorithm and any compression method can be used. Algorithm 5 depicts the generic procedure. The input of the algorithm is a sample $\mathcal{S}$ of time series and an initial guess $z \in \mathcal{T}$. It then repeatedly applies the following steps until termination:

1. A-step: approximate sample mean

$$z \leftarrow \text{AVERAGE}(\mathcal{S}, z).$$

2. C-step: compute a compression chain

$$\mathcal{C}(z) \leftarrow \text{COMPRESS}(z).$$

3. Evaluate solution:

   (a) Select the shortest compression $z_* \in \mathcal{C}(z)$ such that $F(z_*) \leq F(z')$ for all $z' \in \mathcal{C}(z)$.
   (b) If $F(z_*) < F(z)$, set $z \leftarrow z_*$ and go to Step 1, otherwise terminate.

Line 7 computes the complete compression chain $\mathcal{C}(z)$ that consists of all $|z|$ compressions of z of lengths 1 to $|z|$. To accelerate the algorithm at a possible expense of solution quality, sparse compression chains can be considered.

In the following, we explain why and under which conditions compression is useful. To simplify our argument, we assume that AC uses an averaging algorithm for the constrained sample mean problem (such as SSG or DBA). In this case, the length m of the initial guess restricts the search space of AC to the set $\mathcal{T}_{\leq m}$ of all time series of maximum length m. The choice of the length-parameter m via the initial guess is critical. If m is too small, the search space $\mathcal{T}_{\leq m}$ may not contain an unconstrained sample mean. For a given sample $\mathcal{S}$, the Reduction Theorem [22] guarantees the existence of an unconstrained sample mean of a length at most $m_{\mathcal{S}} = \sum_{x \in \mathcal{S}} |x| - 2(|\mathcal{S}| - 1)$. Consequently, we can safely constrain the search space to $\mathcal{T}_{\leq m_{\mathcal{S}}}$ for solving the unconstrained sample mean problem. Then, a naive approach to minimize the Fréchet function on $\mathcal{T}_{\leq m_{\mathcal{S}}}$ is to solve $m_{\mathcal{S}}$ constrained sample mean problems on $\mathcal{T}_{m_{\mathcal{S}}}, \ldots, \mathcal{T}_1$ and then to pick the solution with lowest Fréchet variation. When using state-of-the-art heuristics for the $m_{\mathcal{S}}$ constrained problems, the naive approach is computationally infeasible.

The purpose of compression is to substantially accelerate the intractable naive approach at the expense of solution quality. Instead of solving all $m_{\mathcal{S}}$ constrained problems, the AC

Algorithm 4 MSE Compression

```
1: procedure MSE(x)
2:     initialize compression table C
3:     initialize cost table D such that D[i, j] ← 0 if i = 0 or j = 0
4:     C[1, 1] ← (x_1)
5:     D[1, 1] ← 0
6:     for i ← 1 to |x| do
7:         for m ← 1 to i do
8:             j* ← 0
9:             d* ← ∞
10:            initialize μ* ← [ ]
11:            for j ← m to i do
12:                μ ← (Σ_{l=j}^{i} x_l)/(i − j + 1)
13:                d ← Σ_{l=j}^{i} (x_l − μ)^2
14:                if D[j − 1, m − 1] + d < d* then
15:                    d* ← D[j − 1, m − 1] + d
16:                    μ* ← μ
17:                    j* ← j
18:            D[i, m] ← d*
19:            C[i, m] ← C[j* − 1, m − 1].append(μ*)
20:    C(x) ← {C[|x|, m] : 1 ≤ m ≤ |x|}
21:    return C(x)
```

Algorithm 5 Average-Compress Algorithm

```
1: procedure AC(𝒳, z)
2:     z* ← z          // best solution found so far
3:     f* ← F(z*)      // variation of z*
4:     ℓ* ← |z*|       // length of z*
5:     repeat
6:         A-Step: z ← AVERAGE(𝒳, z)
7:         C-Step: 𝒞(z) ← COMPRESS(z)
8:         //*** Evaluate solution
9:         z ← argmin {F(z') : z' ∈ 𝒞(z) ∪ {z}}
10:        if F(z) < f* or (F(z) = f* and |z| < ℓ*) then
11:            f* ← F(z)
12:            l* ← |z|
13:            z* ← z
14:    until convergence
15:    return z*
```

algorithm uses compressions to select a few promising search spaces $\mathcal{T}_{m_0}, \mathcal{T}_{m_1}, \ldots, \mathcal{T}_{m_k}$ with $m_{\mathcal{S}} = m_0 > m_1 > \cdots > m_k \geq 1$. Starting with $\mathcal{T}_{m_{\mathcal{S}}} = \mathcal{T}_{m_0}$, the solution z_{i-1} found in $\mathcal{T}_{m_{i-1}}$ is compressed in order to determine the next search space $\mathcal{T}_{m_i}$. The length-parameter m_i of the next search space $\mathcal{T}_{m_i}$ corresponds to the length of the compression z_i of z_{i-1} with lowest Fréchet variation. Obviously, this idea only accelerates the naive approach if the length m_i of the best compression is substantially smaller than the length m_{i-1} of the previous solution.

The theoretical upper bound $m_{\mathcal{S}}$ provided by the Reduction Theorem [22] is usually very large such that existing state-of-the-art heuristics for solving the constrained problem on $\mathcal{T}_{\leq m_{\mathcal{S}}}$ are computationally intractable. In this case, also the AC algorithm using such a heuristic would be infeasible. However, empirical results on samples of two time series of equal length n suggest that the length of an unconstrained sample mean is more likely to be less than n [7]. Similar results for larger sample sizes are unavailable due to forbidding running times required for exact sample means. For solving constrained sample mean problems, it is common practice to choose m within the range of the lengths of the sample time series [30]. Within this range, experimental results showed that an approximate solution of a constrained sample mean can be improved by reducing its length using adaptive scaling [30]. These findings suggest to choose the length-parameter m within or slightly above the range of lengths of the sample time series.

3 Experiments

Our goal is to assess the performance of the proposed AC algorithm. For our experiments, we use the 85 data sets from the UCR archive [10]. Appendix 1 summarizes the parameter settings of the mean algorithms used in these experiments.

3.1 Comparison of ADA and MSE

We compared the performance of ADA and MSE as compression subroutines of the AC algorithm. We applied the following configurations:

Acronym	Algorithm	I
DBA	DTW Barycenter Averaging [17, 30]	–
DBA-ADA$_1$	DBA with ADA compression [30]	1
DBA-MSE$_1$	DBA with MSE compression	1
DBA-ADA	DBA with ADA compression	$*$
DBA-MSE	DBA with MSE compression	$*$

Column I refers to the number of iterations of the repeat-until loop of the AC Algorithm. Compression schemes with $*$ iterations run until convergence. We applied DBA and the four AC algorithms to approximate the class means of every UCR training set.[2]

To assess the performance of the mean algorithms, we recorded the percentage deviations, ranking distribution, and space-saving ratios. Here, we used the solutions of the DBA algorithm as reference. The percentage deviation of a mean algorithm A is defined by

$$p_{\text{dev}}(A) = 100 \cdot \frac{F(z_A) - F(z_{\text{DBA}})}{F(z_{\text{DBA}})},$$

where z_{DBA} is the solution of the DBA algorithm and z_A is the solution of algorithm A. Negative (positive) percentage deviations mean that algorithm A has better (worse) Fréchet variation than DBA. The ranking distribution summarizes the rankings of every mean algorithm over all samples. The best (worst) algorithm is ranked first (last). The space-saving ratio of algorithm A is

$$\rho_{\text{ss}}(A) = 1 - \frac{|z_A|}{|z_{\text{DBA}}|}.$$

A positive (negative) space-saving ratio means that the solution z_A is shorter (longer) than z_{DBA}.

Table 1 summarizes the results. The top table shows the average, standard deviation, minimum, and maximum percentage deviations from the Fréchet variation of the DBA algorithm (lower is better). The table in the middle shows the distribution of rankings and their corresponding averages and standard deviations. The best (worst) algorithm is ranked first (fifth). Finally, the bottom table shows the average, standard deviation, minimum, and maximum space-saving ratios (higher is better).

All AC variants improved the solutions of the DBA baseline by 4.6% to 7.0% on average and 45%($\pm$2%) in the best case. By construction, an AC solution is never worse than a DBA solution. The best method is DBA-MSE with average rank 1.0 followed by DBA-MSE$_1$ and DBA-ADA with average ranks 2.4 and 2.5, respectively. These three methods clearly outperformed DBA-ADA$_1$ proposed by Petitjean et al. [30].

On the majority of data sets the AC algorithm converged after two or three iterations to a local optimum. In rare cases, four iterations were required. The number of iterations does neither correlate with the length of the time series nor with the sample size. The main improvement of DBA-MSE and DBA-ADA occurs at the first iteration. The first compression appears to be most aggressive, because it collapses flat regions. Subsequent averaging steps slightly shift the compressed candidate solutions. These shifts may create new flat regions, which can be further compressed to obtain better solutions.

We also considered the lengths of the approximated means. Recall that all mean algorithms started with the same initial guess (medoid). The length of a DBA solution corresponds to

[2] The UCR data sets have prespecified training and test sets.

Table 1 Results of ADA and MSE compressions

Percentage deviations					
	DBA	DBA-ADA$_1$	DBA-MSE$_1$	DBA-ADA	DBA-MSE
avg	0.0	−4.6	−6.0	−5.6	−7.0
std	0.0	4.8	5.9	5.6	6.6
min	0.0	−43.1	−43.9	−47.1	−47.7
max	0.0	0.0	0.0	0.0	0.0
Ranking distribution					
Rank	DBA	DBA-ADA$_1$	DBA-MSE$_1$	DBA-ADA	DBA-MSE
1	0.5	2.5	2.7	10.2	96.8
2	0.2	0.6	56.2	34.6	3.2
3	0.3	9.7	37.0	55.2	0.0
4	0.0	87.1	4.1	0.0	0.0
5	99.0	0.0	0.0	0.0	0.0
avg	5.0	3.8	2.4	2.5	1.0
std	0.3	0.6	0.6	0.7	0.2
Space-saving ratios					
	DBA	DBA-ADA$_1$	DBA-MSE$_1$	DBA-ADA	DBA-MSE
avg	0.00	0.34	0.44	0.36	0.45
std	0.00	0.21	0.24	0.22	0.24
min	0.00	0.00	0.00	0.00	0.00
max	0.00	0.96	0.99	0.96	0.99

the length of its initial guess, whereas solutions of AC algorithms are likely to be shorter by construction. The bottom table shows that solutions of ADA compression save about a third (0.34, 0.36) of the length of DBA solutions on average and solutions of MSE compressions close to a half (0.44, 0.45). As for the Fréchet variation, most of the space-saving occurs in the first iteration of an AC algorithm.

3.2 Comparison of AC algorithms

The goal of the second experiment is to compare the performance of the following mean algorithms:

Acronym	Algorithm	*I*
DBA	DTW Barycenter Averaging [17, 30]	–
SSG	stochastic subgradient method [35]	–
ADBA	adaptive DBA [28]	–
DBA-MSE	DBA with MSE compression	∗
SSG-MSE	SSG with MSE compression	∗
ADBA-MSE	adaptive DBA with MSE compression	∗

Table 2 summarizes the results using the same legend as in Table 1. The percentage deviations and rankings suggest that the three AC variants DBA-MSE, SSG-MSE, and ADBA-MSE performed substantially better than the corresponding base algorithms DBA, SSG, and ADBA, respectively. The SSG-MSE algorithm performed best with an average rank of 1.4, followed by ADBA-MSE (2.9), SSG (3.1), and DBA-MSE (3.3). Interestingly, ADBA performed worse than DBA-MSE. Both, ADBA and DBA-MSE, are based on the DBA algorithm. The difference between both algorithms is that ADBA compresses and expands selected subsequences of the current DBA solution, whereas DBA-MSE only compresses the current DBA solution. The results indicate that simple MSE compression on the entire sequence appears to be a better strategy than ADBA's compression and expansion schemes on selected subsequences. Notably, SSG performed best among the three base averaging algorithms DBA, SSG, ADBA, and performed even better than DBA-MSE. These results are in contrast to those presented in [28], where ADBA outperformed SSG (and also DBA). Our findings confirm that the performance of SSG substantially depends on a careful selection of an optimizer (such as Adam) and a proper choice of the initial learning rate.

Next, we examine the length of the solutions. Note that SSG also does not alter the length of its initial guess such that $\rho_{ss}(\text{DBA}) = \rho_{ss}(\text{SSG}) = 0$. The bottom table shows that MSE compression schemes reduce the length of the solutions obtained by their corresponding base algorithm (DBA, SSG, and ADBA). The space-saving ratios of the AC variants are roughly independent of the particular base algorithm for mean computation (0.43–0.45). Notably, the base algorithm ADBA is more likely to compress rather than to expand the DBA solutions. This finding is in line with the hypothesis that an exact mean is typically shorter than the length of the sample time series [7].

3.3 Qualitative analysis of mean algorithms

The goal of this section is to qualitatively analyze the behavior and phenomena behind the different types of mean algorithms. For this, we considered DBA, ADBA, and the AC variant SSG-MSE relative to an exact dynamic program (EDP) proposed by [7]. Since the sample mean problem is NP-hard, we only considered a sample with two sample time series of length 24 from the Chinatown data set [12]. The two sample time series slightly differ in their amplitudes and on a high abstraction level, they have the following features in common: Both start with a wide valley followed by a peak, a flat plateau-like valley at a high altitude until they finally end with a descent.

Figure 2 shows the sample time series and the sample means returned by the four algorithms. The four algorithms differ in the level of feature abstraction and in susceptibility of spurious features, whereby lower level feature representations are more prone to spurious features than higher level ones: The EDP exhibits the common shape of both sample time series. In addition, it filters out variations of speed by condensing the mean to length 16. The SSG-MSE algorithm more aggressively condenses a solution than EDP resulting in a more compact and higher level description of a mean of length 13. In contrast to EDP, the AC variant SSG-MSE has smoothed out the flat plateau-like valley. We note that this "over-compression" is shape-dependent but not length-dependent. As indicated by Fig. 2, over-compression is more likely for flat and less likely for steep regions. The solution of ADBA more moderately condenses a solution than SSG-MSE and EDP resulting in a lower level representation of length 18. In addition, ADBA includes a spurious plateau at the beginning that occurs only in the upper sample time series. Finally, DBA aims at capturing the common features of both sample time series with respect to a predefined length (here 24). The

Table 2 Results of the AC algorithms

Percentage deviations						
	DBA	SSG	ADBA	DBA-MSE	SSG-MSE	ADBA-MSE
avg	0.0	−7.4	−6.4	−7.0	−11.6	−9.4
std	0.0	16.1	10.4	6.6	13.8	10.5
min	0.0	−79.7	−81.8	−47.7	−83.3	−82.0
max	0.0	188.1	17.7	0.0	50.0	13.3
Ranking distribution						
Rank	DBA	SSG	ADBA	DBA-MSE	SSG-MSE	ADBA-MSE
1	1.9	5.1	1.1	9.5	80.6	8.9
2	2.1	37.6	1.7	22.9	10.5	21.9
3	2.4	20.8	8.6	25.9	2.1	41.4
4	8.1	19.5	33.7	14.0	0.5	21.0
5	11.4	11.1	36.7	27.8	6.3	6.8
6	74.1	5.9	18.3	0.0	0.0	0.0
avg	5.5	3.1	4.6	3.3	1.4	2.9
std	1.1	1.3	1.0	1.3	1.0	1.0
Space-saving ratios						
	DBA	SSG	ADBA	DBA-MSE	SSG-MSE	ADBA-MSE
avg	0.00	0.00	0.31	0.45	0.43	0.45
std	0.00	0.00	0.21	0.24	0.24	0.24
min	0.00	0.00	−0.04	0.00	0.00	0.00
max	0.00	0.00	0.91	0.99	0.99	0.97

resulting solution contains more low level features than the other approaches and includes spurious features which occur in only one of both time series. Finally, we hypothesize that spurious features may also occur in exact solutions when two sample time series do not share many common features.

Figure 2 shows the error of each mean algorithm as the percentage deviation of their Fréchet variations from the minimum Fréchet variation. The Fréchet variation measures the amount of dispersion of a sample of time series. Such a measure is, for example, important in evaluating k-means clustering using validation indices based on the Fréchet variation for each cluster. The errors of the heuristics differ substantially with EDP ranked first followed by SSG-MSE (1.0%), ADBA (10.1%), and DBA (19.3%). We hypothesize that low level and spurious features could result in erroneous measures of dispersion. These errors then propagate to pattern recognition methods based on time series averaging such as k-means clustering.

3.4 Application: *k*-means clustering

In this experiment, we investigated how the quality of a mean algorithm affects the quality of a k-means clustering. Let $\mathcal{S} = \{x_1, \ldots, x_n\} \subseteq \mathcal{T}$ be a set of n finite time series. The goal of

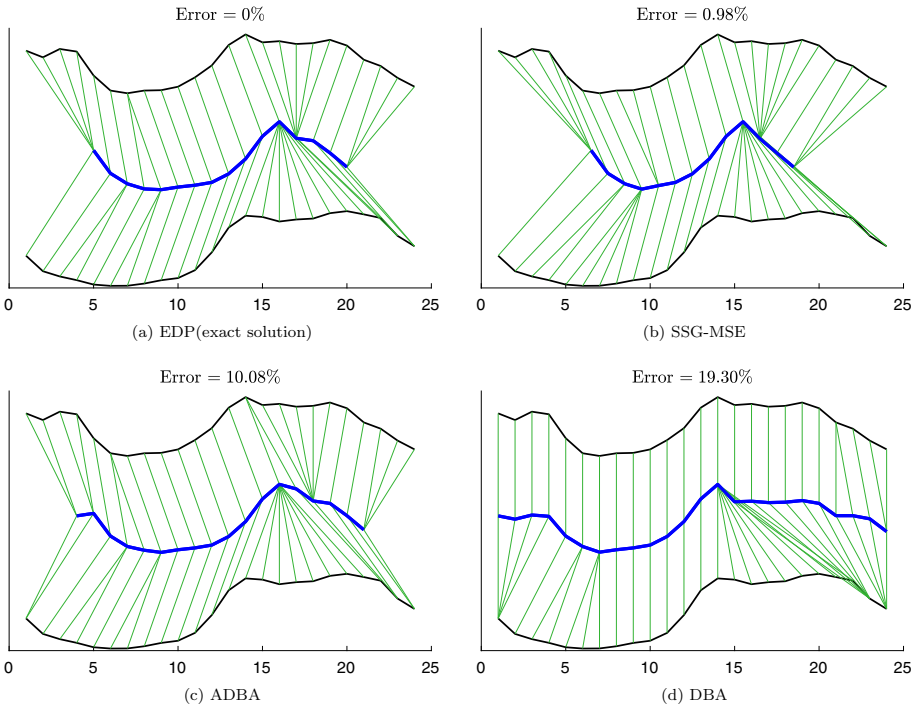


Fig. 2 Comparison of different sample mean algorithms. Each plot shows the mean (blue) of two time series of length 24 (black) and the optimal alignments (green) as found by the corresponding algorithm. The error specified above each plot is the percentage deviation of the Fréchet variation of the corresponding solution from the minimal Fréchet variation. The length of the means are 16 in (**a**), 13 in (**b**), 18 in (**c**), and 24 in (**d**). (Color figure online)

k-means is to find a set $\mathcal{Z} = \{z_1, \ldots, z_k\}$ of k centroids $z_j \in \mathcal{T}$ such that the k-means error

$$J(\mathcal{Z}) = \frac{1}{n} \sum_{i=1}^{n} \min_{z \in \mathcal{Z}} \mathrm{dtw}(x_i, z)^2$$

is minimized. We used DBA, SSG, ADBA, DBA-MSE, SSG-MSE, and ADBA-MSE for computing the set $\mathcal{Z}$ of centroids. We applied the six variants of k-means to 70 UCR data sets and excluded 15 UCR data sets due to overly long running times (see Appendix 1). We merged the prespecified training and test sets. The number k of clusters was set to the number of classes and the centroids were initialized by the class medoids.

Table 3 summarizes the results. The top table presents the average, standard deviation, minimum, and maximum percentage deviations from the respective minimum k-means error (lower is better). The percentage deviation of k-means algorithm A for data set D is defined by

$$p_{\mathrm{dev}}(A, D) = 100 * \frac{J(\mathcal{Z}_A) - J(\mathcal{Z}_D)}{J(\mathcal{Z}_D)},$$

where $\mathcal{Z}_A$ is the set of centroids returned by algorithm A and $\mathcal{Z}_D$ is the best solution obtained by one of the six k-means algorithms. The bottom table shows the distribution of rankings and

Table 3 Results of k-means clustering

Percentage deviations						
	DBA	SSG	ADBA	DBA-MSE	SSG-MSE	ADBA-MSE
Avg	20.0	8.3	8.1	11.5	1.5	3.8
Std	58.1	10.8	4.8	58.1	6.2	3.5
Max	491.8	55.6	26.5	488.5	44.4	17.1
Ranking distribution						
Rank	DBA	SSG	ADBA	DBA-MSE	SSG-MSE	ADBA-MSE
1	2.9	2.9	0.0	14.3	81.4	5.7
2	0.0	22.9	0.0	30.0	12.9	28.6
3	4.3	15.7	14.3	24.3	0.0	42.9
4	7.1	21.4	41.4	11.4	0.0	18.6
5	11.4	31.4	27.1	18.6	5.7	2.9
6	74.3	5.7	17.1	1.4	0.0	1.4
Avg	5.5	3.7	4.5	2.9	1.4	2.9
Std	1.1	1.4	0.9	1.4	1.0	1.0

their corresponding averages and standard deviations. The best (worst) algorithm is ranked first (sixth).

The results show that the AC approach substantially improved all k-means variants using one of the base averaging methods (DBA, SSG, ADBA). Notably, SSG-MSE performed best with an average percentage deviation of 1.5% and an average rank of 1.4, followed by ADBA-MSE (3.8% and 2.9). The average percentage deviations of DBA and DBA-MSE are substantially impacted by the results on a single data set (DiatomSizeReduction). Removing the DiatomSizeReduction data set yields an average percentage deviation of 13.2 for DBA and 4.6 for DBA-MSE, whereas the other average percentage deviations remain unchanged up to $\pm 0.1\%$. These findings confirm the hypothesis raised by Brill et al. [7] that better mean algorithms more likely result in lower k-means errors.

4 Conclusion

We formulated a generic average-compress algorithm for the unconstrained sample mean problem in DTW spaces. Starting with an initial guess of sufficient length, the AC algorithm alternates between averaging and compression. In principle, any averaging and any compression algorithm can be plugged into the AC scheme. The compression guides the algorithm to promising search spaces of shorter time series. This approach is theoretically justified by the Reduction Theorem [22] that guarantees the existence of an unconstrained sample mean in a search space of bounded length. Experimental results show that the AC algorithm substantially outperforms state-of-the-art heuristics for time series averaging. In addition, we observed that better averaging algorithms yield lower k-means errors on average. Open research questions comprise application of the AC scheme to the empirical analysis of alternative compression methods for the AC algorithm and reducing its computational effort.

Funding Open Access funding enabled and organized by Projekt DEAL.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

A Experimental settings

A.1 Hyperparameter settings

In all experiments, we selected the sample medoid as initial guess of a mean algorithm. The DBA algorithm terminated after convergence and latest after 50 epochs (cycles through a sample). The ADBA algorithm terminates subsequence optimization when the sum of the scaling coefficients changes its sign and latest after 50 iterations. The SSG algorithm terminated after 50 iterations without observing an improvement and latest after max $(50, 5000/n)$ epochs. As optimization scheme, SSG applied Adam [27] with $\beta_1 = 0.9$ as first and $\beta_2 = 0.999$ as second momentum. To cope with the problem of selecting an initial learning rate, we used the procedure described in Algorithm 6. The input is a sample $\mathcal{S}$ of size n. The output is the best solution found. The algorithm terminates if the solution did not improve for two consecutive learning rates and latest if $\sqrt{n}/2^i \leq 10^{-6}$.

Algorithm 6 SSG with learning rate selection

```
1: procedure SSG(S)
2:     n ← |S|
3:     i ← 1
4:     repeat
5:         test SSG with learning rate √n/2^i
6:         record best solution z* found so far
7:         i ← i + 1
8:     until convergence
9:     return z*
```

A.2 Data sets excluded from *k*-means experiments

The following list contains all UCR data sets excluded from k-means clustering due to computational reasons:

CinCECGtorso	Phoneme
FordA	StarLightCurves
FordB	UWaveGestureLibraryAll
HandOutlines	UWaveGestureLibraryX
InlineSkate	UWaveGestureLibraryY
Mallat	UWaveGestureLibraryY
NonInvasiveFatalECGThorax1	Yoga
NonInvasiveFatalECGThorax2	

References

1. Abanda, A., Mori, U., Lozano, J.A.: A review on distance based time series classification. Data Min. Knowl. Discov. **33**(2), 378–412 (2019)
2. Abdulla, W.H., Chow, D., Sin, G.: Cross-words reference template for DTW-based speech recognition systems. In: Conference on Convergent Technologies for Asia-Pacific Region, (2003)
3. Aghabozorgi, S., Shirkhorshidi, A.S., Wah, T.-Y.: Time-series clustering—A decade review. Inf. Syst. **53**, 16–38 (2015)
4. 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 Min. Knowl. Discov. **31**(3), 606–660 (2017)
5. Bhattacharya, A., Bhattacharya, R.: Nonparametric Inference on Manifolds with Applications to Shape Spaces. Cambridge University Press, Cambridge (2012)
6. Bellman, R.: On the approximation of curves by line segments using dynamic programming. Commun. ACM **4**(6), 284 (1961)
7. Brill, M., Fluschnik, T., Froese, V., Jain, B., Niedermeier, R., Schultz, D.: Exact mean computation in dynamic time warping spaces. Data Min. Knowl. Discov. **33**(1), 252–291 (2019)
8. Bulteau, L., Froese, V., Niedermeier, R.: Tight hardness results for consensus problems on circular strings and time series. SIAM J. Discrete Math. **34**(3), 1854–1883 (2020)
9. Chakrabarti, K., Keogh, E., Mehrotra, S., Pazzani, M.: Locally adaptive dimensionality reduction for indexing large time series databases. ACM Trans. Database Syst. **27**(2), 188–228 (2002)
10. Chen, Y., Keogh, E., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G.E.: The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/, Accessed: (08/2018)
11. Cuturi, M., Blondel, M.: Soft-DTW: a differentiable loss function for time-series. In: Proceedings of the 34th International Conference on Machine Learning, 70:894–903, (2017)
12. Dau, H.A., Keogh, E., Kamgar, K., Yeh, C.-C.M., Zhu, Y., Gharghabi, S., Ratanamahatana, C.A., Chen, Y., Hu, B., Begum, N., Bagnall, A., Mueen, A., Batista, G. Hexagon, M.L. The UCR Time Series Classification Archive. https://www.cs.ucr.edu/~eamonn/time_series_data_2018/
13. Dryden, I.L., Mardia, K.V.: Statistical Shape Analysis. Wiley, Hoboken (1998)
14. Feragen, A., Lo, P., De Bruijne, M., Nielsen, M., Lauze, F.: Toward a theory of statistical tree-shape analysis. IEEE Trans. Pattern Anal. Mach. Intell. **35**, 2008–2021 (2013)
15. Ferrer, M., Valveny, E., Serratosa, F., Riesen, K., Bunke, H.: Generalized median graph computation by means of graph embedding in vector spaces. Pattern Recognit. **43**(4), 1642–1655 (2010)
16. Fréchet, M.: Les éléments aléatoires de nature quelconque dans un espace distancié. Annales de l'institut Henri Poincaré, 215–310, (1948)
17. Hautamaki, V., Nykanen, P., Franti, P.: Time-series clustering by approximate prototypes. In: International Conference on Pattern Recognition, 1–4, (2008)
18. Huckemann, S., Hotz, T., Munk, A.: Intrinsic shape analysis: geodesic PCA for Riemannian manifolds modulo isometric Lie group actions. Stat. Sin. **20**, 1–100 (2010)
19. Jain, B.: Statistical graph space analysis. Pattern Recognit. **60**, 802–812 (2016)
20. Jain, B., Schultz, D.: Asymmetric learning vector quantization for efficient nearest neighbor classification in dynamic time warping spaces. Pattern Recognit. **76**, 349–366 (2018)
21. Jain, B.: Revisiting inaccuracies of time series averaging under dynamic time warping. Pattern Recognit. Lett. **125**, 418–424 (2019)
22. Jain, B., Schultz, D.: Sufficient conditions for the existence of a sample mean of time series under dynamic time warping. Annals of Mathematics and Artificial Intelligence, 1–34, (2020)
23. Jiang, X., Munger, A., Bunke, H.: On median graphs: properties, algorithms, and applications. IEEE Trans. Pattern Anal. Mach. Intell. **23**(10), 1144–1151 (2001)

24. Kendall, D.G.: Shape manifolds, procrustean metrics, and complex projective spaces. Bull. Lond. Math. Soc. **16**, 81–121 (1984)
25. Keogh, E., Chakrabarti, K., Mehrotra, S., Pazzani, M., Mehrotra, S.: Dimensionality reduction for fast similarity search in large time series databases. Knowl. Inf. Syst. **3**(3), 263–286 (2001)
26. Kim, P.T., Koo, J.-Y.: Comment on [18]. Statistica Sinica **20**, 72–76 (2010)
27. Kingma, D.P., Ba, J.L.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations, (2015)
28. Liu, Y., Zhang, Y., Zeng, M.: Adaptive global time sequence averaging method using dynamic time warping. IEEE Trans. Signal Process. **67**(8), 2129–2142 (2019)
29. Marron, J.S., Alonso, A.M.: Overview of object oriented data analysis. Biom. J. **56**(5), 732–753 (2014)
30. Petitjean, F., Ketterlin, A., Gancarski, P.: A global averaging method for dynamic time warping, with applications to clustering. Pattern Recognit. **44**(3), 678–693 (2011)
31. Petitjean, F., Forestier, G., Webb, G.I., Nicholson, A.E., Chen, Y., Keogh, E.: Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowl. Inf. Syst. **47**(1), 1–26 (2016)
32. Rabiner, L.R., Wilpon, J.G.: Considerations in applying clustering techniques to speaker-independent word recognition. J. Acoust. Soc. Am. **66**(3), 663–673 (1979)
33. Rebagliati, N., Solé, A., Pelillo, M., Serratosa, F.: On The Relation Between The Common Labelling and The Median Graph. Syntactic, and Statistical Pattern Recognition, Structural (2012)
34. Sakoe, H., Chiba, S.: Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans. Acoust. Speech Signal Process. **26**(1), 43–49 (1978)
35. Schultz, D., Jain, B.: Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces. Pattern Recognit. **74**, 340–358 (2018)
36. Soheily-Khah, S., Douzal-Chouakria, A., Gaussier, E.: Generalized k-means-based clustering for temporal data under weighted and kernel time warp. Pattern Recognit. Lett. **75**, 63–69 (2016)
37. Tan, C.W., Webb, G.I., Petitjean, F.: Indexing and classifying gigabytes of time series under time warping. In: Proceedings of the 2017 SIAM International Conference on Data Mining, 282–290, (2017)
38. Terzi, E., Tsaparas, P.: Efficient algorithms for sequence segmentation. In: Proceedings of the 2006 SIAM International Conference on Data Mining, 316–327, (2006)
39. Vintsyuk, T.K.: Speech discrimination by dynamic programming. Cybernetics **4**(1), 52–57 (1968)
40. Wang, H., Marron, J.S.: Object oriented data analysis: sets of trees. Ann. Stat. **35**, 1849–1873 (2007)
41. Wang, H., Song, M.: Ckmeans.1d.dp: optimal k-means clustering in one dimension by dynamic programming. R J. **3**(2), 29–33 (2011)
42. Wilpon, J.G., Rabiner, L.R.: A modified K-means clustering algorithm for use in isolated work recognition. IEEE Trans. Acoust. Speech Signal Process. **33**(3), 587–594 (1985)
43. Yi, B.-Y., Faloutsos, C.: Fast time sequence indexing for arbitrary L_p norms. In: International Conference on Very Large Databases, 385–394, (2000)