Exact mean computation in dynamic time warping spaces

Markus Brill · Till Fluschnik · Vincent Froese · Brijnesh Jain · Rolf Niedermeier · David Schultz

Received: 31 May 2018 / Accepted: 21 November 2018 / Published online: 4 December 2018
© The Author(s), under exclusive licence to Springer Science+Business Media LLC, part of Springer Nature 2018

Abstract
Averaging time series under dynamic time warping is an important tool for improving nearest-neighbor classifiers and formulating centroid-based clustering. The most promising approach poses time series averaging as the problem of minimizing a Fréchet function. Minimizing the Fréchet function is NP-hard and so far solved by several heuristics and inexact strategies. Our contributions are as follows: we first discuss some inaccuracies in the literature on exact mean computation in dynamic time warping spaces. Then we propose an exponential-time dynamic program for computing a global minimum of the Fréchet function. The proposed algorithm is useful for benchmarking and evaluating known heuristics. In addition, we present an exact polynomial-time algorithm for the special case of binary time series. Based on the proposed exponential-time dynamic program, we empirically study properties like uniqueness and length of a mean, which are of interest for devising better heuristics. Experimental evaluations indicate substantial deficits of state-of-the-art heuristics in terms of their output quality.

Keywords Time series analysis · Fréchet function · Dynamic programming · Exact exponential-time algorithm · Empirical evaluation of heuristics

1 Introduction
Time series such as stock prices, weather data, biomedical measurements, and biometrics data (Aghabozorgi et al. 2015; Fu 2011) are time-dependent observations that vary

Responsible editor: Eamonn Keogh.

A short version of this article appeared in the Proceedings of the 2018 SIAM International Conference on Data Mining (SDM ’18), pp. 540–548. SIAM, 2018. This article contains all proofs in full detail. Also, the dynamic program is improved to find an arbitrary weighted mean and new experimental results are included.

Vincent Froese
vincent.froese@tu-berlin.de

Extended author information available on the last page of the article
in length and temporal dynamics. A common technique to deal with these variations in data mining applications is dynamic time warping (dtw) (Sankoff and Kruskal 1999). The dtw-distance has been applied to diverse data mining problems such as speech recognition (Myers and Rabiner 1981), gesture recognition (Alon et al. 2009; Reyes et al. 2011), electrocardiogram frame classification (Huang and Kinsner 2002), and alignment of gene expressions (Aach and Church 2001).

The problem of time series averaging has been subject of research since the 1970s (Abdulla et al. 2003; Cuturi and Blondel 2017; Gupta et al. 1996; Hautamaki et al. 2008; Lummis 1973; Marteau 2018; Petitjean et al. 2011; Rabiner and Wilpon 1979; Schultz and Jain 2018). A common technique to average time series is based on first aligning the time series with respect to the dtw-distance and then synthesizing the aligned time series to an average. Several variations of this approach have been applied to improve nearest-neighbor classifiers and to formulate centroid-based clustering algorithms in dtw-spaces (Rabiner and Wilpon 1979; Oates et al. 1999; Abdulla et al. 2003; Hautamaki et al. 2008; Petitjean et al. 2016; Soheily-Khah et al. 2016). For a detailed overview we refer to Schultz and Jain (2018).

The most successful approaches pose time series averaging as an optimization problem (Cuturi and Blondel 2017; Hautamaki et al. 2008; Petitjean et al. 2011; Schultz and Jain 2018; Soheily-Khah et al. 2016): Suppose that \( \mathcal{X} = (x^{(1)}, \ldots, x^{(k)}) \) is a sample of \( k \) time series \( x^{(i)} \). Then a (weighted Fréchet) mean in dtw-spaces is usually defined as any time series \( z \) that minimizes the weighted Fréchet function (Fréchet 1948)

\[
F_w(z) := \sum_{i=1}^{k} w_i \cdot \text{dtw}(z, x^{(i)})^2,
\]

where \( \text{dtw}(x, y) \) denotes the dtw-distance between time series \( x \) and \( y \) and \( w = (w_1, \ldots, w_k) \) is a given \( k \)-dimensional weight vector (usually assumed to satisfy \( 0 \leq w_i \leq 1 \) for all \( i \in \{1, \ldots, k\} \) and \( \sum_{i=1}^{k} w_i = 1 \)). We refer to the problem of minimizing \( F_w \) over the set \( T \) of all finite time series as the WEIGHTED DTW-MEAN problem. In the special case of \( w_i = 1/k \) for all \( i \in \{1, \ldots, k\} \), we write \( F(z) \) and refer to the problem of minimizing \( F \) over the set \( T \) as DTW-MEAN. Figure 1 shows an example of a mean for two time series. A variant of the DTW-MEAN problem constrains the solution set \( T \) to the subset \( T_m \subseteq T \) of all length-\( m \) time series. For both the constrained and unconstrained DTW-MEAN variant, solutions are guaranteed to exist, but are not unique in general (Schultz and Jain 2018).

As regards relevance of the above objective (the distinguishing feature of a mean \( z \) is that it minimizes the Fréchet variation \( F(z) \) with respect to some distance function), empirical findings suggest that this feature is important for data mining applications. Early approaches to time series averaging such as NLAAF (Gupta et al. 1996) and PSA (Nienattrakul and Ratanamahatana 2009) were algorithmic formulations whose solutions poorly approximated the Fréchet variation (Petitjean et al. 2011; Soheily-Khah et al. 2015). The failure to properly capture the Fréchet variation propagates to poor solutions in \( k \)-means clustering (Petitjean et al. 2011). As a consequence, earlier time series clustering approaches preferred \( k \)-medoids over \( k \)-means (Petitjean et al. 2011; Nienattrakul and Ratanamahatana 2007). The situation changed with
Fig. 1 A mean (blue) of two time series (gray and black) with respect to the dynamic time warping distance (x-axis corresponds to time). Green lines depict the alignment between the mean and the input time series via optimal warping paths (Color figure online)

the advent of principled approaches using an optimization based formulation of time series averaging (Hautamaki et al. 2008; Petitjean et al. 2011). Empirical results show that better approximations of the Fréchet variation result in lower $k$-means loss under dtw-distance (Cuturi et al. 2007; Petitjean et al. 2011) and better performance than $k$-medoids when applied to nearest-neighbor classification (Petitjean et al. 2016). These findings suggest to continue research on devising improved algorithms for minimizing the Fréchet function.

As has been shown very recently, DTW-Mean is NP-hard, W[1]-hard with respect to the parameter sample size $k$, and not solvable in $\rho(k) \cdot n^{o(k)}$ time for any computable function $\rho$ (where $n$ is the maximum length of any input time series) assuming a plausible complexity-theoretic hypothesis (Bulteau et al. 2018). Exponential-time algorithms based on averaging global multiple (or $k$-dimensional) alignments have been falsely claimed to provide optimal solutions (Hautamaki et al. 2008; Petitjean and Gançarski 2012; Petitjean et al. 2011). Existing heuristics approximately solve the constrained DTW-Mean problem (Cuturi and Blondel 2017; Petitjean et al. 2011; Schultz and Jain 2018) in which the length $m$ of feasible solutions is specified beforehand without any knowledge about whether the subset $T_m$ contains an optimal solution of the unconstrained DTW-Mean problem. In summary, the development of nontrivial exact algorithms is to be considered widely open.

Our contributions We discuss several problematic statements in the literature concerning the computational complexity of exact algorithms for DTW-Mean. We refute (supplying counterexamples) some false claims from the literature and clarify the known state of the art with respect to computing means in dtw-spaces (Sect. 3). We show that in case of binary time series (both input and mean) there is an exact polynomial-time algorithm for mean computation in dtw-spaces (Sect. 4). We develop a dynamic program as an exact algorithm for the (unconstrained) Weighted DTW-Mean problem on rational time series. The worst-case time complexity of the proposed dynamic program is $O(n^{2^k+1}k^k)$, where $k$ is the number of input time series and $n$ is the maximum length of a sample time series (Sect. 5). We apply the proposed exact dynamic program on small-scaled problems as a benchmark of how well the state-of-the-art heuristics approximate a mean. Our empirical findings indicate that all state-of-the-art heuristics suffer from relatively poor worst-case solution quality in terms of minimizing the Fréchet function, and the solution quality in general may vary quite a lot (Sect. 6).


2 Preliminaries

Throughout this paper, we consider only finite univariate time series with rational elements. A univariate time series of length \( n \) is a sequence \( x = (x_1, \ldots, x_n) \in \mathbb{Q}^n \). We denote the set of all univariate rational time series of length \( n \) by \( T_n \). Furthermore, \( T = \bigcup_{n \in \mathbb{N}} T_n \) denotes the set of all univariate rational time series of finite length. For every \( n \in \mathbb{N} \), let \([n]\) denote the set \( \{1, \ldots, n\} \).

The next definition is fundamental for our central computational problem.

Definition 1 A warping path of order \( m \times n \) is a sequence \( p = (p_1, \ldots, p_L) \) with \( p_\ell \in [m] \times [n] \) for all \( \ell \in [L] \) such that

(i) \( p_1 = (1, 1) \),
(ii) \( p_L = (m, n) \), and
(iii) \( p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\} \) for all \( \ell \in [L-1] \).

Note that \( \max\{m, n\} \leq L \leq m + n \). We denote the set of all warping paths of order \( m \times n \) by \( \mathcal{P}_{m,n} \). For two time series \( x = (x_1, \ldots, x_m) \) and \( y = (y_1, \ldots, y_n) \), a warping path \( p = (p_1, \ldots, p_L) \in \mathcal{P}_{m,n} \) defines an alignment of \( x \) and \( y \): each pair \( p_\ell = (i_\ell, j_\ell) \) of \( p \) aligns element \( x_{i_\ell} \) with \( y_{j_\ell} \). The cost \( C_p(x, y) \) for aligning \( x \) and \( y \) along warping path \( p \) is defined as \( C_p(x, y) = \sum_{\ell=1}^L (x_{i_\ell} - y_{j_\ell})^2 \). The dtw-distance between \( x \) and \( y \) is defined as

\[
\text{dtw}(x, y) := \min_{p \in \mathcal{P}_{m,n}} \left\{ \sqrt{C_p(x, y)} \right\}.
\]

A warping path \( p \) with \( C_p(x, y) = (\text{dtw}(x, y))^2 \) is called an optimal warping path for \( x \) and \( y \).

We remark that \( \text{dtw}(x, y) \) for two time series \( x, y \) of length \( n \) can be computed in subquadratic time \( O(n^2 \log \log n / \log \log n) \) (Gold and Sharir 2018). The existence of a strongly subquadratic-time (that is, \( O(n^{2-\epsilon}) \) for some \( \epsilon > 0 \)) algorithm is considered unlikely (Bringmann and Künnemann 2015).

Our central computational problem is defined as follows.

**Weighted DTW-Mean**

**Input:** Sample \( \mathcal{X} = (x^{(1)}, \ldots, x^{(k)}) \) of \( k \) univariate rational time series and rational nonnegative weights \( w_1, \ldots, w_k \).

**Task:** A univariate rational time series \( z \) that minimizes the weighted Fréchet function \( F_w(z) \).

The special case of uniform weights (that is, \( w_i = \frac{1}{k} \) for all \( i \in [k] \)) is called DTW-Mean. It is known that a weighted mean always exists (Jain and Schultz 2018, Remark 2.13) (however, it is not necessarily unique). Note that for rational inputs, every weighted mean is also rational (see Lemma 5), that is, Weighted DTW-Mean always has a solution.
3 Problematic statements in the literature

In this section we discuss misleading and wrong claims in the literature.

3.1 NP-hardness

The DTW-MEAN problem is often related to the STEINER STRING (STS) problem (Hautamaki et al. 2008; Petitjean et al. 2011, 2016; Petitjean and Gańcarski 2012; Aghabozorgi et al. 2015; Fahiman et al. 2017; Paparrizos and Gravano 2017; Marteau 2018). A Steiner string (Gusfield 1997) (or Steiner sequence) for a set $S$ of strings is a string $t$ that minimizes $\sum_{s \in S} D(t, s)$, where $D$ is a distance measure between two strings often assumed to fulfill the triangle inequality (e.g., the weighted edit distance). Computing a Steiner string is equivalent to solving the MULTIPLE SEQUENCE ALIGNMENT (MSA) problem (Gusfield 1997). Both, STS and MSA, are known to be NP-hard for several distance measures even on binary alphabets (Bonizzoni and Della Vedova 2001; Nicolas and Rivals 2005). Interestingly, as we show in Sect. 4, WEIGHTED DTW-MEAN is solvable in polynomial time for the space of binary time series.

Several papers mention the NP-hardness results for MSA and STS in the context of DTW-MEAN (Hautamaki et al. 2008; Marteau 2018; Paparrizos and Gravano 2017; Fahiman et al. 2017), thereby suggesting that DTW-MEAN is also NP-hard. However, it is not clear (and not shown in the literature) how to reduce from MSA (or STS) to DTW-MEAN since the involved distance measures are significantly different. For example, the dtw-distance lacks the metric property of the edit distance. Very recently, devising a reduction from the MULTICOLORED CLIQUE problem, Bulteau et al. (2018) showed that DTW-MEAN is NP-hard, W[1]-hard with respect to the sample size $k$, and not solvable in time $\rho(k) \cdot n^{o(k)}$ for any function $\rho$, where $n$ is the maximum length of any sample time series (unless the Exponential Time Hypothesis fails).

3.2 Computation of exact solutions

Two exponential-time algorithms were proposed to exactly solve DTW-MEAN. The first approach is based on multiple (sequence) alignment in bioinformatics (Gusfield 1997) and the second is a “brute-force” method (Petitjean et al. 2011, Section 3.2). We show that neither algorithm is guaranteed to return an optimal solution for every DTW-MEAN instance.

Multiple alignment It is claimed that DTW-MEAN can be solved by averaging a (global) multiple alignment of the $k$ input series (Hautamaki et al. 2008; Petitjean et al. 2011, 2016; Petitjean and Gańcarski 2012). A multiple alignment of $k$ time series in the context of dynamic time warping is described as computing a $k$-dimensional warping path in a $k$-dimensional matrix (a precise formal definition is not given). Concerning the running time, it is claimed that computing a multiple alignment requires $\Theta(n^k)$ time (Petitjean et al. 2011; Petitjean and Gańcarski 2012) $[O(n^k)$ time (Petitjean et al. 2011; Petitjean and Gańcarski 2012) $]$. This statement is misleading because the running time is significantly higher due to the exponential nature of the problem.

The second approach is the brute-force method, which involves computing all possible alignments and selecting the one with the minimum cost. This method is computationally expensive, especially for large datasets, but it guarantees the optimal solution. However, the statement that this method is a “brute-force” approach is misleading because it does not emphasize the specific steps involved in the algorithm. The method is actually an exhaustive search, which is known to be exponential in the worst case. The statement that this method runs in $O(n^k)$ time is misleading as well, because the actual running time is much higher due to the combinatorial explosion of alignments.

Thus, both methods have their limitations, and it is important to clarify the assumptions and the actual complexity of the algorithms when discussing the NP-hardness and the computation of exact solutions for DTW-MEAN.
where \( n \) is the maximum length of an input time series. Neither the upper bound of \( O(n^k) \) nor the lower bound of \( \Omega(n^k) \) on the running time are formally proven.

Given a multiple alignment, it is claimed that averaging the \( k \) resulting aligned time series column-wise yields a mean (Petitjean and Gańcarski 2012, Definition 4). We show that this is not correct even for \( k = 2 \) time series where a multiple alignment is simply obtained by an optimal warping path. However, the column-wise average of two aligned time series obtained from an optimal warping path is not always an optimal solution for a DTW-MEAN instance as the following example shows.

**Example 1** Let \( x^{(1)} = (1, 4, 2, 3) \) and \( x^{(2)} = (4, 2, 4, 5) \). By exhaustive search, we found the unique optimal warping path for \( x^{(1)} \) and \( x^{(2)} \)

\[
p = ((1, 1), (2, 1), (3, 2), (4, 3), (4, 4))
\]

of length five. The two corresponding aligned length-five time series are

\[
x_p^{(1)} = (1, 4, 2, 3, 3),
\]
\[
x_p^{(2)} = (4, 4, 2, 4, 5).
\]

The arithmetic mean of these time series is \( \bar{x} = (2.5, 4, 2, 3.5, 4) \). However, for \( z = (2.5, 4, 2, 4) \), we have \( F(z) = 3.25 < 3.5 = F(\bar{x}) \), which shows that \( \bar{x} \) is not a mean (see Fig. 2).

In Example 1, the time series \( \bar{x} \) is also not an optimal choice among all time series of length five since also \( z' = (2.5, 4, 4, 2, 4) \) satisfies \( F(z') = 3.25 < F(\bar{x}) \). In fact, by computer-based exhaustive search we found that no warping path \( p \in \mathcal{P}_{4,4} \) yields a mean for \( x^{(1)} \) and \( x^{(2)} \) by averaging the aligned time series \( x_p^{(1)} \) and \( x_p^{(2)} \). We conclude that a multiple alignment as defined by Petitjean and Gańcarski (2012, Definition 5) that shall produce an averaged time series that minimizes the Fréchet function does not exist in general. Example 1 implies that incremental pairwise averaging strategies such as NLAAF (Gupta et al. 1996) or PSA (Nienattrakul and Ratanamahatana 2009) are based on inexact mean computation for two time series.

We finish with another erroneous example from the literature for three time series (Petitjean and Gańcarski 2012, Figure 2).
**Example 2** For the three time series

\[
\begin{align*}
x^{(1)} &= (1, 10, 0, 0, 4), \\
x^{(2)} &= (0, 2, 10, 0, 0), \\
x^{(3)} &= (0, 0, 10, 0, 0),
\end{align*}
\]

the multiple alignment is given as

\[
\begin{align*}
x^{(1)'} &= (1, 1, 1, 10, 0, 0, 4), \\
x^{(2)'} &= (0, 0, 2, 10, 0, 0, 0), \\
x^{(3)'} &= (0, 0, 0, 10, 0, 0, 0),
\end{align*}
\]

yielding the arithmetic mean \( \bar{x} = \left( \frac{1}{3}, \frac{1}{3}, 1, 10, 0, 0, \frac{4}{3} \right) \) with \( F(\bar{x}) = 14/3 \geq 4.66 \). However, for \( z = \left( \frac{1}{4}, 1, 10, 0, 0, \frac{4}{3} \right) \), it holds \( F(z) \leq 4.48 < F(\bar{x}) \).

**Search algorithm** Another approach to solve DTW-MEAN is based on searching through potential solutions (Petitjean et al. 2011, Section 3.2). Suppose an optimal mean is of length \( m \). Consider for each input time series a partition into \( m \) consecutive non-empty parts and align the \( i \)th element in the mean with all elements in the \( i \)th part of each time series. It is claimed that a mean can be found by trying out all possible partitions into \( m \) consecutive non-empty parts for each input time series. This approach is not correct since not all possible solutions are considered (as two elements in the mean can be aligned with the same element of an input time series). Example 1 depicts this problem.

**4 Polynomial-time solvability for binary data**

By restricting the values in the time series (input and mean) to be binary (0 or 1), we arrive at the special case **Binary Weighted DTW-MEAN**.

**Binary Weighted DTW-MEAN**

**Input:** Sample \( \mathcal{X} = (x^{(1)}, \ldots, x^{(k)}) \) of \( k \) time series with elements in \( \{0, 1\} \) and rational nonnegative weights \( w_1, \ldots, w_k \).

**Task:** Find a time series \( z \in \{0, 1\}^* \) that minimizes \( F_w(z) \).

We prove that **Binary Weighted DTW-MEAN** is polynomial-time solvable.

**Theorem 1** Binary Weighted DTW-MEAN for \( k \) input time series is solvable with \( O(kn^3) \) arithmetic operations, where \( n \) is the maximum length of any input time series.

The underlying idea is to show that a binary mean can always be assumed to have length \( O(n) \) and that it does not contain two equal consecutive entries. Hence, there are only \( O(n) \) possible binary candidates to test. We first prove some preliminary results about the dtw-distance of binary time series and properties of a binary mean. We start with the following general definition.
Definition 2 A time series \( x = (x_1, \ldots, x_n) \) is condensed if no two consecutive elements are equal, that is, \( x_i \neq x_{i+1} \) holds for all \( i \in [n-1] \). We denote the condensation of a time series \( x \) by \( \tilde{x} \) and define it to be the time series obtained by repeatedly removing one of two equal consecutive elements in \( x \) until the remaining series is condensed.

The following proposition implies that a mean can always be assumed to be condensed. Note that this holds for arbitrary time series (not only for the binary case) but not for the constrained mean.

Proposition 1 Let \( x \) be a time series and let \( \tilde{x} \) denote its condensation. Then, for every time series \( y \), it holds that \( \text{dtw}(\tilde{x}, y) \leq \text{dtw}(x, y) \).

Proof Let \( y \) have length \( m \) and assume that \( x = (x_1, \ldots, x_n) \) is not condensed. Then, \( x_i = x_{i+1} \) holds for some \( i \in [n-1] \). Let \( p = ((i_1, j_1), \ldots, (i_L, j_L)) \) be an optimal warping path for \( x \) and \( y \). Now, consider the time series \( x' = (x_1, \ldots, x_i, x_{i+2}, \ldots, x_n) \) that is obtained from \( x \) by deleting the element \( x_{i+1} \). We construct a warping path \( p' \) for \( x' \) and \( y \) such that \( C_{p'}(x', y) = C_p(x, y) \). To this end, let \( p_a = (i_a, j_a), 2 \leq a \leq L \), be the first index pair in \( p \) where \( i_a = i + 1 \) (hence, \( i_{a-1} = i \)). Now, we consider two cases.

If \( j_a = j_{a-1} + 1 \), then we define the order-\((n-1) \times m)\) warping path

\[
p' := ((i_1, j_1), \ldots, (i_{a-1}, j_{a-1}), (i_a - 1, j_a), \ldots, (i_L - 1, j_L))
\]

of length \( L \). Thus, each element of \( y \) that was aligned to \( x_{i+1} \) in \( p \) is now aligned to \( x_i \) instead. To check that \( p' \) is a valid warping path, note first that \( (i_1, j_1) = (1, 1) \) and \( (i_L - 1, j_L) = (n - 1, m) \) holds since \( p \) is a warping path. Also, it holds

\[
\forall 1 \leq \ell \leq a - 2: \quad (i_{\ell+1}, j_{\ell+1}) - (i_\ell, j_\ell) \in \{(1, 0), (0, 1), (1, 1)\},
\]

\[
(i_a - 1, j_a) - (i_{a-1}, j_{a-1}) = (0, 1),
\]

\[
\forall a \leq \ell \leq L - 1: \quad (i_{\ell+1} - 1, j_{\ell+1}) - (i_\ell - 1, j_\ell) \in \{(1, 0), (0, 1), (1, 1)\}.
\]

The cost of \( p' \) is

\[
C_{p'}(x', y) = \sum_{\ell=1}^{a-1} (x'_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a}^{L} (x'_{(i_\ell-1)} - y_{j_\ell})^2
\]

\[
= \sum_{\ell=1}^{a-1} (x_{i_\ell} - y_{j_\ell})^2 + \sum_{\ell=a}^{L} (x_{(i_\ell-1)} - y_{j_\ell})^2 = C_p(x, y).
\]

Otherwise, if \( j_a = j_{a-1} \), then we define the warping path

\[
p' := ((i_1, j_1), \ldots, (i_{a-1}, j_{a-1}), (i_{a+1} - 1, j_{a+1}), \ldots, (i_L - 1, j_L))
\]

of length \( L - 1 \). Again, each element of \( y \) that was aligned to \( x_{i+1} \) in \( p \) is now aligned to \( x_i \) instead and it holds \( (i_1, j_1) = (1, 1) \) and \( (i_L - 1, j_L) = (n - 1, m) \) since \( p \) is a
warping path. Clearly, also
\[
\forall 1 \leq \ell \leq a - 2: \quad (i_{\ell+1}, j_{\ell+1}) - (i_{\ell}, j_{\ell}) \in \{(1, 0), (0, 1), (1, 1)\} \text{ and }
\forall a + 1 \leq \ell \leq L - 1: \quad (i_{\ell+1} - 1, j_{\ell+1}) - (i_{\ell} - 1, j_{\ell}) \in \{(1, 0), (0, 1), (1, 1)\}
\]
holds. Finally, we have \((i_{a+1} - 1, j_{a+1}) - (i_{a-1}, j_{a-1}) \in \{(1, 0), (0, 1), (1, 1)\}\) since
\(i_{a+1} - 1 - i_{a-1} = i_{a+1} - i_a\) holds and also \(j_{a+1} - j_{a-1} = j_{a+1} - j_a\) holds. Thus, \(p'\)
is a valid warping path and its cost is
\[
C_{p'}(x', y) = \sum_{\ell=1}^{a-1} (x'_{i_{\ell}} - y'_{j_{\ell}})^2 + \sum_{\ell=a+1}^{L} (x'_{i_{\ell} - 1} - y'_{j_{\ell}})^2
= \sum_{\ell=1}^{a-1} (x_{i_{\ell}} - y_{j_{\ell}})^2 + \sum_{\ell=a+1}^{L} (x_{i_{\ell}} - y_{j_{\ell}})^2
= C_p(x, y) - (x_{i_a} - y_{j_a})^2.
\]
Since in both cases above, the cost does not increase, we obtain
\[
dtw(x', y) \leq C_{p'}(x', y) \leq C_p(x, y) = dtw(x, y).
\]
Repeating this argument until \(x'\) is condensed finishes the proof. \(\square\)

Proposition 1 implies that we can assume a mean to be condensed. Next, we want
to prove an upper bound on the length of a binary mean. To this end, we analyze
the \(dtw\)-distances of binary time series. Note that a binary condensed time series is fully
determined by its first element and its length. We use this property to give a closed
expression for the \(dtw\)-distance of two condensed binary time series.

**Lemma 1** Let \(x = (x_1, \ldots, x_n), \ y = (y_1, \ldots, y_m) \in \{0, 1\}^*\) be two condensed
binary time series with \(n \geq m\). Then, it holds
\[
dtw(x, y)^2 = \begin{cases} 
\lfloor (n - m)/2 \rfloor, & x_1 = y_1 \\
2, & x_1 \neq y_1 \land n = m \\
1 + \lfloor (n - m)/2 \rfloor, & x_1 \neq y_1 \land n > m
\end{cases}
\]

**Proof** We prove the statement by first giving a warping path that has the claimed cost
and second proving that every warping path has at least the claimed cost.

“\(\leq\)”: We show that there exists a warping path \(p\) between \(x\) and \(y\) that has the
claimed cost. The warping path \(p\) is defined as follows:
If \(x_1 = y_1\), then we have \(x_i = y_i\) for all \(i \in [m]\) (since \(x\) and \(y\) are condensed and
binary) and we set
\[
p := ((1, 1), (2, 2), \ldots, (m, m), (m + 1, m), \ldots, (n, m)).
\]
This warping path has cost \(C_p(x, y) = \sum_{i=m+1}^{n} |x_i - y_m| = \lfloor (n - m)/2 \rfloor\).
If \( x_1 \neq y_1 \), then we have \( x_i = y_{i-1} \) for all \( 2 \leq i \leq m \). Thus, for \( n = m \), the warping path

\[
p := ((1, 1), (2, 1), (3, 2), \ldots, (n, m - 1), (n, m))
\]

has cost \( C_p(x, y) = \left| x_1 - y_1 \right| + \left| x_n - y_m \right| = 2 \). Finally, for \( n > m \), the warping path \( p := ((1, 1), (2, 1), (3, 2), \ldots, (m + 1, m), (m + 2, m), \ldots, (n, m)) \) yields cost

\[
C_p(x, y) = \left| x_1 - y_1 \right| + \sum_{i=2}^{m+1} \left| x_i - y_{i-1} \right| + \sum_{i=m+2}^{n} \left| x_i - y_m \right|
\]

\[
= 1 + \sum_{i=m+2}^{n} \left| x_i - y_m \right| = 1 + \left\lceil \frac{n - m - 1}{2} \right\rceil = 1 + \left\lceil \frac{(n - m)}{2} \right\rceil.
\]

“\( \geq \)” We show that every warping path has at least the cost claimed above. Consider an optimal warping path \( p = (p_1, \ldots, p_L) \) for \( x \) and \( y \) and note that there are at least \( n - m \) different indices \( \ell \in [L - 1] \) such that \( p_{\ell+1} - p_\ell = (1, 0) \) since \( n \geq m \). For every such pairs \( p_{\ell+1} = (i_{\ell+1}, j_{\ell+1}), p_\ell = (i_\ell, j_\ell) \) with \( j_{\ell+1} = j_\ell \), we have

\[
\left| x_{i_{\ell+1}} - y_{j_{\ell+1}} \right| + \left| x_{i_\ell} - y_{j_\ell} \right| = \left| x_{i_{\ell+1}} - y_{j_\ell} \right| + \left| x_{i_\ell} - y_{j_{\ell+1}} \right| = 1
\]

since \( x_{j_{\ell+1}} \neq x_{j_\ell} \) (recall that \( x \) is condensed and binary). Hence, at least for every second such index \( \ell \) (starting from the first one) a cost of 1 is induced. Hence,

\[
dtw(x, y)^2 \geq \left\lceil \frac{(n - m)}{2} \right\rceil.
\]

If \( x_1 = y_1 \), then this lower bound matches the claimed cost.

If \( x_1 \neq y_1 \) and \( n = m \), then also \( x_n \neq y_m \) and hence the cost is at least 2.

If \( x_1 \neq y_1 \) and \( n > m \), then we can assume that \( p_2 = (2, 1) \). To see this, note that for \( p_2 \neq (2, 1) \) the cost is at least \( 1 + \left\lceil \frac{(n - m)}{2} \right\rceil \) by the above argument. If \( p_2 = (2, 1) \), then the subpath \((p_2, p_3, \ldots, p_L)\) of \( p \) is an optimal warping path between \((x_2, \ldots, x_n)\) and \( y \), where \( x_2 = y_1 \) and \( n - 1 \geq m \). As we have already shown above, this path has cost \( \left\lceil \frac{(n - 1 - m)}{2} \right\rceil \). Hence, in this case \( p \) has cost \( 1 + \left\lceil \frac{(n - 1 - m)}{2} \right\rceil = 1 + \left\lceil \frac{(n - m)}{2} \right\rceil \). Thus, we can assume that \( p_2 = (2, 1) \) in which case the cost of \( p \) matches the claimed cost of the lemma. This finishes the proof.

Note that according to Lemma 1, for a fixed condensed binary time series \( y \) of length \( m \), the value \( \text{dtw}(x, y)^2 \) is monotonically increasing in the length of \( x \) for all condensed binary time series \( x \) of length \( n \geq m + 1 \). We use this property later in the proof of Lemma 3 where we derive an upper bound on the length of a binary mean. In order to prove Lemma 3, we also need the following lemma concerning the dtw-distances between condensed and non-condensed time series.

**Lemma 2** Let \( x = (x_1, \ldots, x_n) \) be a condensed binary time series and let \( y = (y_1, \ldots, y_m) \in \{0, 1\}^* \) with \( n \geq m \). Then, for the condensation \( \tilde{y} \) of \( y \) it holds

\[
\text{dtw}(x, y)^2 = \text{dtw}(x, \tilde{y})^2.
\]
Proof Assume that \( y \) is not condensed. Then, \( y \) consists of \( \ell \in [m] \) blocks, where a block is a maximal subsequence of consecutive 0’s or consecutive 1’s in \( y \). Let \( m_1, \ldots, m_\ell \) denote the lengths of these blocks where \( m_1 + \ldots + m_\ell = m \). Note also that \( \tilde{y} \) has length \( \ell \) with \( \ell < m \leq n \). We define a warping path \( p \) between \( x \) and \( y \) such that \( C_p(x, y) = \text{dtw}(x, \tilde{y})^2 \). Note that, by Lemma 1, we have

\[
\text{dtw}(x, \tilde{y})^2 = \begin{cases} 
(n - \ell)/2, & x_1 = \tilde{y}_1 \\
1 + (n - \ell)/2, & x_1 \neq \tilde{y}_1 
\end{cases}
\]

If \( x_1 = y_1 \), then we set \( p := ((1, 1), \ldots, (1, m_1), (2, m_1 + 1), \ldots, (2, m_1 + m_2), \ldots, (\ell, m), (\ell + 1, m), \ldots, (n, m)) \) and obtain cost \( C_p(x, y) = \sum_{i=\ell+1}^{n} |x_i - y_m| = [(n - \ell)/2] \).

If \( x_1 \neq y_1 \), then we set \( p := ((1, 1), (2, 1), \ldots, (2, m_1), (3, m_1 + 1), \ldots, (3, m_1 + m_2), \ldots, (\ell + 1, m), (\ell + 2, m), \ldots, (n, m)) \) and obtain cost

\[
C_p(x, y) = 1 + \sum_{i=\ell+2}^{n} |x_i - y_m| = 1 + [(n - \ell)/2].
\]

\( \square \)

We now have all ingredients to show that there always exists a binary mean of length at most one larger than the maximum length of any input time series.

Lemma 3 For binary input time series \( x^{(1)}, \ldots, x^{(k)} \in \{0, 1\}^* \) of maximum length \( n \), there exists a binary mean \( z \in \{0, 1\}^* \) of length at most \( n + 1 \).

Proof Assume that \( z = (z_1, \ldots, z_m) \in \{0, 1\}^* \) is a mean of length \( m > n + 1 \). By Proposition 1, we can assume that \( z \) is condensed, that is, \( z_i \neq z_{i+1} \) for all \( i \in [m-1] \).

We claim that \( z' := (z_1, \ldots, z_{n+1}) \) is also a mean. We prove this claim by showing that \( \text{dtw}(z, x^{(i)})^2 \leq \text{dtw}(z', x^{(i)})^2 \) holds for all \( i \in [k] \). By Lemmas 1 and 2, we have \( \text{dtw}(z', x^{(i)})^2 = \text{dtw}(z', \tilde{x}^{(i)})^2 \leq \text{dtw}(z, \tilde{x}^{(i)})^2 \leq \text{dtw}(z, x^{(i)})^2 \), where the inequality follows from Lemma 1 since \( z' \) is of length \( n + 1 < m \) and the dtw-distance is monotonically increasing.

Having established that a binary mean can always be assumed to be condensed and of bounded length, we now show that it can be found in polynomial time.

Proof of Theorem 1 By Proposition 1 and Lemma 3, we can assume the desired mean \( z \in \{0, 1\}^* \) to be a condensed series of length at most \( n + 1 \). Thus, there are at most \( 2n + 2 \) many possible candidates for \( z \). For each candidate \( z \), we can compute the value \( F_w(z) \) with \( O(kn^2) \) arithmetic operations and select the one with the smallest value. Overall, this yields \( O(kn^3) \) arithmetic operations.

\( \square \)

5 An exact algorithm solving DTW-MEAN

We develop a nontrivial exponential-time algorithm solving WEIGHTED DTW-MEAN exactly. The key is to observe a certain structure of a mean and the corresponding
alignments to the input time series. To this end, we define redundant elements in a mean. Note that this concept was already used by Jain and Schultz (2018, Theorem 2.7) in order to prove the existence of a mean of bounded length [though (Jain and Schultz 2018 Definition 3.20) is slightly different].

**Definition 3** Let \( x^{(1)}, \ldots, x^{(k)} \) and \( z = (z_1, \ldots, z_m) \) be time series and let \( p^{(j)}, j \in [k] \), denote an optimal warping path between \( x^{(j)} \) and \( z \). We call an element \( z_i \) of \( z \) **redundant** if in every time series \( x^{(j)} \), \( j \in [k] \), there exists an element that is aligned with \( z_i \) and with another element of \( z \) by \( p^{(j)} \).

The next lemma states that there always exists a mean without redundant elements (similarly to Jain and Schultz 2018, Theorem 2.7).

**Lemma 4** There exist a mean \( z \) for time series \( x^{(1)}, \ldots, x^{(k)} \) and optimal warping paths \( p^{(j)} \) between \( z \) and \( x^{(j)} \) for each \( j \in [k] \) such that \( z \) contains no redundant element.

**Proof** Let \( z \) be a mean of \( x^{(1)}, \ldots, x^{(k)} \) with optimal warping paths \( p^{(j)} \), \( j \in [k] \), such that the element \( z_i \) is redundant (recall that a mean \( z \) always exists). We show that there also exists a mean \( z' \) and optimal warping paths \( p^{(j)'} \) such that no element in \( z' \) is redundant.

Assume first that there exists a \( j \in [k] \) such that the element \( z_i \) is aligned by \( p^{(j)} \) with at least one element \( x^{(j)}_t \) in \( x^{(j)} \) that is not aligned with any other element in \( z \). Let \( L \) denote the length of \( p^{(j)} \). Then, \( p^{(j)} \) is of the form

\[
p^{(j)} = (p_1, \ldots, (i - 1, \ell_{t-1}), (i, \ell_t), \ldots, (i, \ell_{t+\alpha}), (i + 1, \ell_{t+\alpha+1}), \ldots, p_L)
\]

for some \( \ell_t \leq \ell \leq \ell_{t+\alpha} \) and \( \alpha \geq 1 \). Since \( z_i \) is redundant, it follows that \( \ell_{t-1} = \ell_t \) or \( \ell_{t+\alpha} = \ell_{t+\alpha+1} \) holds. If \( \ell_{t-1} = \ell_t \), then we remove the pair \((i, \ell_t)\) from \( p^{(j)} \). Also, if \( \ell_{t+\alpha} = \ell_{t+\alpha+1} \), then we remove the pair \((i, \ell_{t+\alpha})\) from \( p^{(j)} \). Note that this yields a warping path \( p^{(j)'} \) between \( z \) and \( x^{(j)} \) since even if we removed both pairs \((i, \ell_t)\) and \((i, \ell_{t+\alpha})\), then we know by assumption that there still exists the pair \((i, \ell)\) with \( \ell_t \leq \ell \leq \ell_{t+\alpha} \) in \( p^{(j)'} \) since \( z_i \) is aligned with \( x^{(j)}_t \) which is not aligned with another element in \( z \). Since we only removed pairs from \( p^{(j)} \), it holds \( C_{p^{(j)'}(z, x^{(j)})} \leq C_{p^{(j)}(z, x^{(j)})} \). Moreover, \( z_i \) is not redundant anymore.

Now, assume that for all \( j \in [k] \), \( z_i \) is aligned only with elements in \( x^{(j)} \) which are also aligned with another element of \( z \) by \( p^{(j)} \) (that is, \( z_i \) is redundant according to Jain and Schultz 2018, Definition 3.20). Let \( z' \) denote the time series obtained by deleting the element \( z_i \) from \( z \). Jain and Schultz (2018, Proof of Theorem 2.7) showed that in this case \( dtw(z', x^{(j)}) \leq dtw(z, x^{(j)}) \) holds for all \( j \in [k] \) (the basic idea is that removing \( z_i \) yields valid warping paths where some pairs are removed, which gives a smaller cost).

Under both assumptions above, we reduced the number of redundant elements. Hence, we can repeat the above arguments until we obtain a mean \( z' \) without redundant elements.
Lemma 4 allows us to devise a dynamic program computing a mean without redundant elements. We compute a mean by testing all possibilities to align the last mean element to elements from the input time series while recursively adding an optimal solution for the remaining non-aligned elements in the input time series (see Fig. 3). We use the assumption that the mean does not contain redundant elements for this recursive approach.

Before describing our dynamic program, we prove the following lemma concerning the optimal value of a mean element for given alignments.

**Lemma 5** Let $x^{(1)}, \ldots, x^{(k)}$ be time series and let $w_1, \ldots, w_k$ be nonnegative weights. Let $n_j$ denote the length of $x^{(j)}$, $j \in [k]$. Further, let $p^{(j)}$ be a warping path of order $m \times n_j$ for $m \in \mathbb{N}$ and let $z = \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^{k} w_j C_{p^{(j)}}(x, x^{(j)})$. For $i \in [m]$, let $x^{(j)}_{\ell^{(j)}_i}, \ldots, x^{(j)}_{h^{(j)}_i}$ denote the elements of $x^{(j)}$ that are aligned with element $z_i$ by $p^{(j)}$. Then,

$$z_i = \frac{\sum_{j=1}^{k} w_j \sum_{t=\ell^{(j)}_i}^{h^{(j)}_i} x^{(j)}_t}{\sum_{j=1}^{k} w_j (h^{(j)}_i - \ell^{(j)}_i + 1)}.$$

**Proof** By the assumption of the lemma, we have

$$z = \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^{k} w_j C_{p^{(j)}}(x, x^{(j)}) = \arg \min_{x \in \mathcal{T}_m} \sum_{j=1}^{k} w_j \sum_{i=1}^{m} \sum_{t=\ell^{(j)}_i}^{h^{(j)}_i} (x_i - x^{(j)}_t)^2$$

$$= \arg \min_{x \in \mathcal{T}_m} \sum_{i=1}^{m} \sum_{j=1}^{k} w_j \sum_{t=\ell^{(j)}_i}^{h^{(j)}_i} (x_i - x^{(j)}_t)^2.$$
Hence, for each $i \in [m]$, it holds

$$z_i = \arg \min_{\mu \in \mathbb{Q}} \sum_{j=1}^{k} w_j \sum_{t=\ell_{ij}}^{h_{ij}} (\mu - x_{i_j}^{(j)})^2.$$  

Note that the above sum is a convex function in $\mu$ (since all weights are nonnegative). Setting the first derivative with respect to $\mu$ equal to zero yields

$$\sum_{j=1}^{k} \left( 2w_j(h_{ij} - \ell_{ij} + 1)z_i - 2w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_{i_j}^{(j)} \right) = 0$$

$$\iff 2z_i \sum_{j=1}^{k} (w_j(h_{ij} - \ell_{ij} + 1)) = 2\sum_{j=1}^{k} w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_{i_j}^{(j)}$$

$$\iff z_i = \frac{\sum_{j=1}^{k} w_j \sum_{t=\ell_{ij}}^{h_{ij}} x_{i_j}^{(j)}}{\sum_{j=1}^{k} w_j(h_{ij} - \ell_{ij} + 1)}$$  

\[\square\]

We now prove our main theorem.

**Theorem 2** **Weighted DTW-Mean** for $k$ input time series is solvable with $O(n^{2k+1}2^k k)$ arithmetical operations, where $n$ is the maximum length of any input time series.

**Proof** Assume for simplicity that all time series have length $n$ (the general case can be solved analogously). We find a mean using a dynamic programming approach. Let $C$ be a $k$-dimensional table, where for all $(i_1, \ldots, i_k) \in [n]^k$, we define

$$C[i_1, \ldots, i_k] = \min_{z \in T} \left( \sum_{j=1}^{k} w_j \left( \text{dtw}(z, (x_{i_1}^{(j)}, \ldots, x_{i_k}^{(j)})) \right)^2 \right),$$

that is, $C[i_1, \ldots, i_k]$ is the value $F_w(z)$ of the weighted Fréchet function of a mean $z$ for the subseries $(x_{i_1}^{(1)}, \ldots, x_{i_k}^{(1)}), \ldots, (x_{i_1}^{(k)}, \ldots, x_{i_k}^{(k)})$. Clearly, $C[n, \ldots, n]$ is the optimal value $F_w(z)$ of a mean for the input instance.

For $i_1 = i_2 = \cdots = i_k = 1$, a mean $z$ clearly contains just one element and each optimal warping path between $z$ and $(x_{i_1}^{(j)})$ trivially equals $((1, 1))$. By Lemma 5, we initialize

$$C[1, \ldots, 1] = \sum_{j=1}^{k} w_j (x_{i_1}^{(j)} - \mu)^2, \quad \mu = \frac{\sum_{j=1}^{k} w_j x_{i_1}^{(j)}}{k}.$$  

Hence, the corresponding mean is $z = (\mu)$.  

© Springer
For the case that \(i_j > 1\) holds for at least one \(j \in [k]\), consider a mean \(z\) for \((x^{(1)}_1, \ldots, x^{(1)}_{i_1}), \ldots, (x^{(k)}_1, \ldots, x^{(k)}_{i_k})\). By Lemma 4, we can assume that there exist optimal warping paths \(p^{(j)}\) between \(z\) and \((x^{(j)}_1, \ldots, x^{(j)}_{i_j})\) such that \(z\) contains no redundant elements. Let \(z_q\) be the last element of \(z\). Then, for each \(j \in [k]\), \(z_q\) is aligned by \(p^{(j)}\) with some elements \(x^{(j)}_{\ell_j}, \ldots, x^{(j)}_{i_j}\) for \(\ell_j \in [i_j]\). By Lemma 5, it follows that

\[
z_q = \frac{\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x^{(j)}_t}{\sum_{j=1}^k w_j (i_j - \ell_j + 1)}.
\]

Hence, the contribution of \(z_q\) to \(F_w(z)\) is \(\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x^{(j)}_t - z_q)^2\).

Now, assume that there exists another element \(z_{q-1}\) in \(z\). Clearly, for each \(j \in [k]\), \(z_{q-1}\) is aligned only with elements of indices up to \(\ell_j\) since otherwise the warping path conditions are violated. Hence, \(F_w(z)\) can be obtained recursively from a mean of the subseries \((x^{(1)}_1, \ldots, x^{(1)}_{\ell_1}), \ldots, (x^{(k)}_1, \ldots, x^{(k)}_{\ell_k})\). Recall, however, that we assumed \(z\) not to contain any redundant element. It follows that \(z_{q-1}\) cannot be aligned with \(x^{(j)}_{\ell_j}\) for all \(j \in [k]\) since \(z_q\) is already aligned with each \(x^{(j)}_{\ell_j}\). Hence, we add the minimum value \(C[\ell'_1, \ldots, \ell'_k]\) over all \(\ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}\), where \(\ell'_j = \ell_j - 1\) holds for at least one \(j \in [k]\).

We arrive at the following recursion:

\[
C[i_1, \ldots, i_k] = \min \{ c^*(\ell_1, \ldots, \ell_k) + \sigma(\ell_1, \ldots, \ell_k) \mid \ell_1 \in [i_1], \ldots, \ell_k \in [i_k] \},
\]

where

\[
\sigma(\ell_1, \ldots, \ell_k) := \sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x^{(j)}_t - \mu)^2, \quad \mu := \frac{\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x^{(j)}_t}{\sum_{j=1}^k w_j (i_j - \ell_j + 1)},
\]

and

\[
c^*(\ell_1, \ldots, \ell_k) := \min \left\{ C[\ell'_1, \ldots, \ell'_k] \mid \ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}, \sum_{j=1}^k (\ell_j - \ell'_j) > 0 \right\}
\]

if \(\ell_j > 1\) holds for some \(j \in [k]\), and \(c^*(1, \ldots, 1) := 0\).

In order to compute \(C[i_1, \ldots, i_k]\), a minimum is computed over all possible choices \(\ell_j \in [i_j], j \in [k]\). For each choice \(\ell_1, \ldots, \ell_k\), the value \(\mu\) corresponds to the last element of a mean and \(\sigma(\ell_1, \ldots, \ell_k)\) is the induced cost of aligning this element with \(x^{(j)}_{\ell_j}, \ldots, x^{(j)}_{i_j}\) for each \(j \in [k]\). The value \(c^*(\ell_1, \ldots, \ell_k)\) yields the value \(F_w(z')\) of a mean \(z'\) for the remaining subseries \((x^{(j)}_{\ell_j'}, \ldots, x^{(j)}_{i_j'})\), \(j \in [k]\), over all \(\ell'_j \in \{\max(1, \ell_j - 1), \ell_j\}\) such that \(\sum_{j=1}^k (\ell_j - \ell'_j) > 0\), which implies
Algorithm 1: Exact Dynamic Program (EDP) for WEIGHTED DTW-MEAN

Input: Time series $x^{(1)}, \ldots, x^{(k)}$ of length $n$ and weights $w_1, \ldots, w_k > 0$.
Output: Mean $z$ and $F_w(z)$.
Initialize $C$ \hspace{1cm} // $k$-dimensional DP table storing $F_w$-values
Initialize $Z$ \hspace{1cm} // $k$-dimensional table storing means

foreach $(i_1, \ldots, i_k) \in [n]^k$ do \hspace{1cm} // fill tables iteratively
\hspace{1cm} $C[i_1, \ldots, i_k] := \infty$
\hspace{1cm} $Z[i_1, \ldots, i_k] := ()$

foreach $(\ell_1, \ldots, \ell_k) \in [i_1] \times \cdots \times [i_k]$ do \hspace{1cm} // compute $C[i_1, \ldots, i_k]$
\hspace{1cm} $\mu := (\sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} x_t^{(j)})/(\sum_{j=1}^k w_j (i_j - \ell_j + 1))$
\hspace{1cm} $\sigma := \sum_{j=1}^k w_j \sum_{t=\ell_j}^{i_j} (x_t^{(j)} - \mu)^2$
\hspace{1cm} $c^* := \infty$
\hspace{1cm} $z := ()$
\hspace{1cm} if $\ell_1 = \ell_2 = \cdots = \ell_k = 1$ then
\hspace{1.5cm} $c^* := 0$
\hspace{1cm} else \hspace{1cm} // compute $c^*(\ell_1, \ldots, \ell_k)$ based on table look-ups
\hspace{1.5cm} foreach $(\ell'_1, \ldots, \ell'_k) \in [\ell_1 - 1, \ell_1] \times \cdots \times [\ell_k - 1, \ell_k]$ do
\hspace{1.7cm} if $\forall j \in [k] : \ell'_j \geq 1$ and $\exists j \in [k] : \ell'_j < \ell_j$ then
\hspace{2cm} if $C[\ell'_1, \ldots, \ell'_k] < c^*$ then
\hspace{2.5cm} $c^* := C[\ell'_1, \ldots, \ell'_k]$
\hspace{2cm} $z := Z[\ell'_1, \ldots, \ell'_k]$
\hspace{1.7cm} if $c^* + \sigma < C[i_1, \ldots, i_k]$ then \hspace{1cm} // update mean and $F_w$-value
\hspace{2cm} $C[i_1, \ldots, i_k] := c^* + \sigma$
\hspace{2cm} $Z[i_1, \ldots, i_k] := \text{append}(z, \mu)$
\hspace{1cm} return $(Z[n, \ldots, n], C[n, \ldots, n])$

that $\ell'_j = \ell_j - 1$ holds for at least one $j \in [k]$. This condition guarantees that we only find alignments which do not yield redundant elements in the mean (which we can assume by Lemma 4). Note that $\ell_j = 1$ implies that $\ell'_j = 1$ (since index 0 does not exist in $x^{(j)}$).

The dynamic programming table $C$ can be filled iteratively along the dimensions starting from $C[1, \ldots, 1]$. The overall number of entries is $n^k$. For each table entry, the minimum of a set containing $O(n^k)$ elements is computed. Computing an element requires the computation of $\sigma(\ell_1, \ldots, \ell_k)$ which can be done with $O(kn)$ arithmetical operations plus the computation of $c^*(\ell_1, \ldots, \ell_k)$ which is the minimum of a set of size at most $2^k$ whose elements can be obtained by constant-time table look-ups. Thus, $C$ can be filled using $O(n^k \cdot n^k \cdot 2^k \cdot kn)$ arithmetical operations. A mean can be obtained by storing the values $\mu$ for which the minimum in the above recursion is attained (Algorithm 1 contains the pseudocode).

We close with some remarks on the above result.

- The dynamic program also allows to compute all (non-redundant) means by storing all possible values for which the minimum in the recursion is attained.
– It is possible to incorporate a fixed length \( q \) into the dynamic program (by adding another dimension to the table \( C \)) such that it outputs only optimal solutions of length \( q \).\(^1\) The running time increases by a factor of \( q^2 \).
– The dynamic program can easily be extended to multivariate time series with elements in \( \mathbb{Q}^d \) and a cost function \( C_p(x, y) := \sum_{\ell=1}^{L} \| x_{i\ell} - y_{j\ell} \|_2^2 \) (with a running time increase by a factor of \( d \)).

### 6 Experiments

The goal of this section is twofold: first, we empirically study some characteristics of a mean which might be of interest for devising heuristics as well as for mean-based applications in data mining. Second, we assess the performance of state-of-the-art heuristics in terms of minimizing the Fréchet function.

#### 6.1 Test data

We used data sets derived from random walks and the 27 UCR data sets (Chen et al. 2015) listed in Table 1. Due to the prohibitive running time of Algorithm 1, our experiments are limited to data sets with a small number of short time series. Note that this limitation is due to the intrinsic hardness of DTW-Mean [(presumably there is no algorithm running faster than \( n^{ck} \) for some constant \( c > 0 \) (Bulteau et al. 2018)].

Random walks were used to conduct controlled experiments within the same problem domain in order to investigate mean properties and to assess the performance of heuristics more objectively under different conditions.

A random walk \( x = (x_1, \ldots, x_n) \) is of the form

\[
x_1 = \varepsilon_1, \\
x_i = x_{i-1} + \varepsilon_i \quad \text{for all } 2 \leq i \leq n,
\]

where the \( \varepsilon_i \) are random numbers drawn from the normal distribution \( N(0, 1) \). For the UCR data sets we merged the training and test sets. Time series within each UCR data set have the same length \( n \). We restricted the experiments to data sets whose time series have length \( n \leq 150 \).

Unless otherwise stated, we generated samples according to the following procedures:

– \( S_{rw} \): For every \( n \in \{10, 20, \ldots, 100\} \), we generated 1000 pairs of random walks of length \( n \) giving a total of 10,000 samples of size \( k = 2 \).
– \( S_{rw}^k \): For every \( k \in \{2, \ldots, 6\} \), we generated 1000 samples consisting of \( k \) random walks of length \( n = 6 \) giving a total of 5000 samples.
– \( S_{ucr} \): For every UCR data set, we randomly sampled 1000 different pairs of time series giving a total of 27,000 samples of size \( k = 2 \).

\(^1\) Source code available at [http://www.akt.tu-berlin.de/menue/software/](http://www.akt.tu-berlin.de/menue/software/).
Table 1  List of 27 UCR time series data sets

| Data Set                | #    | n   | Type          |
|-------------------------|------|-----|---------------|
| ItalyPowerDemand        | 1096 | 24  | SENSOR        |
| SyntheticControl        | 600  | 60  | SIMULATED     |
| SonyAIBORobotSurface2   | 980  | 65  | SENSOR        |
| SonyAIBORobotSurface1   | 621  | 70  | SENSOR        |
| ProximalPhalanxTW       | 605  | 80  | IMAGE         |
| ProximalPhalanxOutlineCorrect | 891 | 80 | IMAGE         |
| ProximalPhalanxOutlineAgeGroup | 605 | 80 | IMAGE         |
| PhalangesOutlinesCorrect | 2658 | 80 | IMAGE         |
| MiddlePhalanxTW         | 553  | 80  | IMAGE         |
| MiddlePhalanxOutlineCorrect | 891 | 80 | IMAGE         |
| MiddlePhalanxOutlineAgeGroup | 554 | 80 | IMAGE         |
| DistalPhalanxTW         | 539  | 80  | IMAGE         |
| DistalPhalanxOutlineCorrect | 876 | 80 | IMAGE         |
| DistalPhalanxOutlineAgeGroup | 539 | 80 | IMAGE         |
| TwoLeadECG              | 1162 | 82  | ECG           |
| MoteStrain              | 1272 | 84  | SENSOR        |
| ECG200                  | 200  | 96  | ECG           |
| MedicalImages           | 1141 | 99  | IMAGE         |
| TwoPatterns             | 5000 | 128 | SIMULATED     |
| SwedishLeaf             | 1125 | 128 | IMAGE         |
| CBF                     | 930  | 128 | SIMULATED     |
| FacesUCR                | 2250 | 131 | IMAGE         |
| FaceAll                 | 2250 | 131 | IMAGE         |
| ECGFiveDays             | 884  | 136 | ECG           |
| ECG5000                 | 5000 | 140 | ECG           |
| Plane                   | 210  | 144 | SENSOR        |
| GunPoint                | 200  | 150 | MOTION        |

Columns # and n show the number and length of time series, respectively. The last column refers to the respective application domains (e.g. ECG stands for electrocardiography)

6.2 Mean properties

The first set of experiments studies mean properties which are of interest for devising better heuristics and mean-based applications. Due to the high computational costs of finding exact means, the majority of experiments focused on properties of condensed means. The analysis of properties of arbitrary means are confined to a subset of tiny scale problems.

6.2.1 Uniqueness

Non-unique means can cause problems in theory and practice. For example, proving that sample means are consistent estimators of population means becomes more
complicated for non-unique means. Furthermore, non-unique means can introduce undesired ambiguities into mean-based applications. For example, \(k\)-means clustering in dtw-spaces minimizes a non-convex cost function by applying the EM-algorithm (Hautamaki et al. 2008; Petitjean et al. 2014, 2016; Soheily-Khah et al. 2016). For non-convex problems, the performance of the EM-algorithm depends on the initial starting point (Moon 1996). The means obtained in the maximization-step of one iteration of \(k\)-means are the initial centroids of the next iteration. Hence, we hypothesize that the performance of \(k\)-means clustering in dtw-spaces (Hautamaki et al. 2008; Petitjean et al. 2014, 2016; Soheily-Khah et al. 2016) does not only depend on the choice of initial means as demonstrated by Petitjean et al. (2011) but also on the choice of recomputed means during optimization. In this case, the extent of these difficulties depends—at least in principle—on the prevalence of non-unique means. In the following, we investigate the prevalence of non-unique condensed (and non-condensed) means.

**Setup** We applied Algorithm 1 to all samples of type \(S_{rw}\), \(S_{rk}\), and \(S_{ucr}\). The percentage \(P_{ucm}\) of unique condensed means as well as the average and maximum number of condensed means, denoted by \(\bar{\alpha}\) and \(\alpha^*\), were recorded. For samples consisting of pairs of time series of length \(n \leq 40\), we tested the existence of non-condensed means. These samples are of type \(S_{rw}\) with \(k = 2\) and \(n \in \{10, 20, 30, 40\}\) and the 1000 samples from the *ItalyPowerDemand* data set.

**Results and discussion** Table 2 summarizes the results. Figure 4 shows the estimated cumulative distribution function of the number \(\alpha\) of condensed means over all 42,000 samples. We made the following observations:

1. A condensed mean is unique for the majority of samples (see column \(P_{ucm}\) of Table 2 and Fig. 4).
2. Non-condensed means occur exceptionally, that is, in less than 2.5% of all samples with short time series (see column \(P_{ncm}\) of Table 2).
3. The average and maximum number of condensed means is between one and two for all but four UCR data sets (see columns \(\bar{\alpha}\) and \(\alpha^*\) of Table 2).

These findings indicate that unique (condensed) means are more likely than non-unique means. The implications of the proposed observations are twofold: first, mean-based methods such as \(k\)-means clustering are less likely to be prone to potential problems caused by ambiguities. Second, the observations give rise to the hypothesis that a (condensed) mean of a sample is unique *almost everywhere* in a measure-theoretic sense. If this hypothesis is true, then the aforementioned problems caused by non-unique means are hopefully negligible in practice [we remark that optimal warping paths are unique almost everywhere (Jain and Schultz 2017) which might indicate that the above hypothesis holds].

A notable exception is the (simulated) *Two_Patterns* data set with only 0.5% of samples with a unique condensed mean. Figure 5 shows two typical time series of this data set and describes why their condensed mean is not uniquely determined. From the description it follows that warping of two constant plateaus can cause non-
Table 2 The table on the top shows properties on uniqueness and length of condensed means

| Condensed means | Data set                          | \( P_{\text{ucm}} \) | \( \bar{\alpha} \) | \( \alpha^* \) | \( \bar{\delta} \) |
|-----------------|-----------------------------------|------------------------|---------------------|-----------------|---------------------|
| \( S_{rw} \)    | 100.0                             | 1.0 (± 0.00)           | 1.0                 | -30.3 (± 25.79)  |
| \( S_{kw} \)    | 100.0                             | 1.0 (± 0.00)           | 1.0                 | -38.5 (± 18.91)  |
| ItalyPowerDemand| 99.4                              | 1.0 (± 0.08)           | 2.0                 | -5.8 (± 11.29)   |
| Synthetic_control| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -8.3 (± 16.87)   |
| SonyAIBORobotSurface| 43.9                             | 2.2 (± 2.01)          | 16.0                | -13.9 (± 5.53)   |
| SonyAIBORobotSurfaceII| 48.2                             | 2.1 (± 2.03)          | 16.0                | -12.9 (± 7.15)   |
| ProximalPhalanxTW| 80.2                              | 1.2 (± 0.40)           | 2.0                 | -16.0 (± 13.79)  |
| ProximalPhalanxOutlineCorrect| 69.5                             | 1.3 (± 0.46)           | 2.0                 | -19.9 (± 13.48)  |
| ProximalPhalanxOutlineAgeGroup| 78.6                             | 1.2 (± 0.41)           | 2.0                 | -17.5 (± 13.15)  |
| PhalangesOutlines| 82.5                              | 1.2 (± 0.38)           | 2.0                 | -17.3 (± 13.24)  |
| MiddlePhalanxTW| 82.8                              | 1.2 (± 0.38)           | 2.0                 | -15.6 (± 13.67)  |
| MiddlePhalanxOutlineCorrect| 88.3                             | 1.1 (± 0.32)           | 2.0                 | -13.8 (± 13.76)  |
| MiddlePhalanxOutlineAgeGroup| 85.2                             | 1.1 (± 0.36)           | 2.0                 | -14.7 (± 13.64)  |
| DistalPhalanxTW| 74.5                              | 1.3 (± 0.44)           | 2.0                 | -17.9 (± 13.20)  |
| DistalPhalanxOutlineCorrect| 81.7                             | 1.2 (± 0.39)           | 2.0                 | -16.4 (± 13.46)  |
| DistalPhalanxOutlineAgeGroup| 76.6                             | 1.2 (± 0.42)           | 2.0                 | -17.6 (± 13.21)  |
| TwoLeadECG| 94.7                              | 1.1 (± 0.26)           | 4.0                 | -9.3 (± 8.07)    |
| MoteStrain| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -30.9 (± 16.55)  |
| ECG200| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -8.8 (± 9.59)    |
| MedicalImages| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -32.1 (± 14.34)  |
| Two_Patterns| 0.5                               | 7.9 (± 4.66)           | 16.0                | -27.3 (± 6.40)   |
| SwedishLeaf| 100.0                             | 1.0 (± 0.00)           | 1.0                 | 0.9 (± 6.13)     |
| CBF| 100.0                             | 1.0 (± 0.00)           | 1.0                 | 2.8 (± 6.82)     |
| FacesUCR| 99.7                              | 1.0 (± 0.05)           | 2.0                 | -8.2 (± 5.85)    |
| FaceAll| 99.8                              | 1.0 (± 0.04)           | 2.0                 | -8.1 (± 5.72)    |
| ECGFiveDays| 97.4                              | 1.0 (± 0.16)           | 2.0                 | -22.3 (± 13.87)  |
| ECG5000| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -21.5 (± 16.04)  |
| Plane| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -2.3 (± 4.68)    |
| Gun_Point| 100.0                             | 1.0 (± 0.00)           | 1.0                 | -49.1 (± 14.02)  |

| Non-condensed means | Data set | \( P_{\text{nrcm}} \) |
|---------------------|----------|----------------------|
| \( S_{rw} \) with \( 10 \leq n \leq 40 \) |          | 0.3                  |
| ItalyPowerDemand    |          | 2.1                  |

Notation

\( P_{\text{ucm}} \) Percentage of unique condensed means
\( P_{\text{nrcm}} \) Percentage of non-condensed means
\( \bar{\alpha} \) Average number of condensed means
Table 2 continued

| Notation | Description |
|----------|-------------|
| $\alpha^*$ | Maximum number of condensed means |
| $\delta$ | Average length-deviation |

Numbers in parentheses show standard deviations. The table below shows the percentage of non-condensed means for a restricted set of samples. The bottom table describes the notation used as column identifiers.

![Fig. 4 Estimated cumulative distribution function (cdf) of the number $\alpha$ of condensed means over all 42,000 samples. The right plot shows an excerpt of the left plot with the cdf restricted to the interval $[0.85, 1]$](image)

Uniqueness. Such plateaus are common in the Two_Patterns data set which explains the low percentage of unique condensed means.

6.2.2 Length

For a given sample, the current state-of-the-art heuristics (Cuturi and Blondel 2017; Hautamaki et al. 2008; Petitjean et al. 2011; Schultz and Jain 2018) approximate a mean by approximately solving a constrained DTW-MEAN problem. The constrained problem restricts the set of feasible solutions to the subset $T_m$ of time series of length $m$. The parameter $m$ is typically chosen within the range of the lengths of the sample time series. The two main questions are whether the subset $T_m$ contains a mean at all and what the best possible approximation we can achieve is when constraining the solution set to $T_m$? We start with the first question.

**Setup** We applied Algorithm 1 to all samples of type $S_{rw}$, $S^k_{rw}$, and $S_{ucr}$. For every sample $\mathcal{X}$, the set of all condensed means was computed and the lengths were recorded. We computed the length-deviation

$$\delta_{\mathcal{X}} := 100 \cdot \frac{m_{cm} - n}{n},$$

where $n$ is the length of the sample time series and $m_{cm}$ is the average length of all condensed means of $\mathcal{X}$. Negative (positive) values of $\delta_{\mathcal{X}}$ mean that $m_{cm}$ is $|\delta_{\mathcal{X}}|$ percent smaller (larger) than $n$. 
Results and discussion

Table 2 and Fig. 6 summarize the results. To discuss the results, we consider the constrained Fréchet variation

$$F^*_m = \min_{z \in T_m} F(z)$$

for every length $m \in \mathbb{N}$.

The first observation to be made is that different condensed means of the same sample have the same length for all 42,000 samples. Recall that non-condensed means occur only rarely (see column $P_{ncm}$ of Table 2). Thus, means of a sample $X$ are typically condensed with identical length $m_{cm}$. Under these conditions, the function $F^*_m$ has a unique minimum at $m_{cm}$. Thus, for a majority of samples, the solution space $T_m$ does

---

**Fig. 5** Two time series $x^{(1)}$ and $x^{(2)}$ from the Two_Patterns data set with eight different condensed means. 
(a) An example of a mean and corresponding optimal warping paths between the mean and both sample time series. The three colored warpings show regions that cause non-uniqueness. 
(b), (c) A detailed view of the green warpings. In (b) the mean of (a) is depicted. In (c) another mean is shown. In (b) and (c) both plateaus of $x^{(1)}$ and $x^{(2)}$ are aligned with only two points in the mean; one point is aligned with all but one points of the plateau of $x^{(1)}$, the other point is aligned with all but one points of the plateau of $x^{(2)}$. This is optimal because both plateaus are completely flat (constant). The only difference in (b) and (c) is the order of the two mean points. Each pair of warped plateaus provides two possibilities to construct a mean. Consequently, the three warped pairs of plateaus in (a) induce $2^3 = 8$ condensed means (Color figure online)
Fig. 6 Average length $m$ of condensed means (red lines) as functions of the length of sample time series $n$ (left) and sample size $k$ (right). The gray shaded areas depict the standard deviations and the dashed red lines the minimum and maximum length of condensed means. The black lines show the length of the sample time series (Color figure online)

not contain a mean for any $m \neq m_{\text{cm}}$. This gives rise to the question of how the length $m_{\text{cm}}$ of condensed means is related to the length $n$ of the sample time series. We observed the following:

1. The average length of condensed means is (substantially) less than the length of the sample time series for all but two UCR data sets (see column $\delta$ of Table 2).
2. The average length $m_{\text{cm}}$ of condensed means increases with increasing length $n$ of random walks [see Fig. 6 (left)]. The best linear fit in a squared error sense has slope 0.58 indicating that $m_{\text{cm}}$ increases slower than $n$. No correlation between average length-deviations and length of sample time series could be observed across different UCR data sets.
3. The average length $m_{\text{cm}}$ of condensed means decreases with increasing sample size $k$ of random walks [see Fig. 6 (right)].

The above observations have the following implications: most state-of-the-art heuristics were tested on sample time series with identical length $n$ using $m = n$ as predefined mean length (Cuturi and Blondel 2017; Petitjean et al. 2011; Schultz and Jain 2018). The empirical findings suggest that the length of a condensed mean is substantially shorter than the standard setting $m = n$. Consequently, in most cases, the solution space $\mathcal{T}_n$ of state-of-the-art methods does not contain a mean. Thus, setting the mean length equal to the length of the input time series introduces a structural error that can not be overcome by any solver of the corresponding constrained DTW-MEAN problem. The results suggest to consider mean lengths which are smaller than $n$. Note that in this case the computation time of mean-algorithms should decrease.

We now address the second main question, that is, how the choice of a mean length $m$ affects the structural error $\varepsilon_m := F^*_m - F^*_s$, where $F^*_m = \min_{z \in \mathcal{T}_m} F(z)$ is the constrained and $F^*_s = \min_{z \in \mathcal{T}} F(z)$ is the unconstrained Fréchet variation. To estimate the structural error, we study the constrained Fréchet variation $F^*_m$ as a function of $m$.

Setup For every $n \in \{5, 10, 15, \ldots, 40\}$ we generated 1000 pairs of random walks. We applied Algorithm 1 to all samples and computed the constrained Fréchet variations $F^*_m$ for all $m \in \{1, \ldots, 60\}$. 

$\square$ Springer
Results and discussion

Figures 7 and 8 depict the main characteristics of the functions $F_m^*$ as functions of length $m \in \{1, \ldots, 60\}$ shown in lin-log scale. The functions $F_m^*$ were computed for 1000 samples consisting of two random walks of length $n = 40$. The plot depicts 20 randomly selected functions $F_m^*$. The highlighted red line is the average function $F_m^*$ over all 1000 functions (Color figure online).

We observed the following typical behavior: The function $F_m^*$ first rapidly decreases on $D_1$ until it reaches its global minimum at $m_{cm}$ and then increases on $D_2$ with a small slope. The shape of $F_m^*$ roughly resembles an exponential decay on $D_1$ followed by a linear tail on $D_2$. Figure 8a categorizes all observed shapes of $F_m^*$ on $D_1$ and $D_2$. The most common shape of $F_m^*$ is strictly monotonous on $D_1$ (96.0%) and $D_2$ (98.2%).

Figure 8b presents a typical example of $F_m^*$. Non-monotonous or constant parts in a curve occur only rarely. These findings indicate that the function $F_m^*$ is strictly convex in most cases. This implies that the further away $m$ is from the global minimum $m_{cm}$, the larger the structural error $\varepsilon_m$ is.

Table 3 summarizes the slopes of the best linear fits of $F_m^*$ on $D_2$ and the error percentages $E = 100 \cdot (F_m^* - F_n)/F_n$ of the constrained Fréchet variation at parameter $m = n$, where $n$ is the length of the sample time series. The median slopes on $D_2$ are low $a_{med} \approx 0.0034$ for $n = 5$ and tend to further decrease with increasing length $n$ to $a_{med} \approx 0.0001$ for $n = 40$. As a consequence of the low median slopes, the median error percentages decline from $E_{med} \approx 0.42$ to less than $E_{med} \approx 0.01$ for increasing $n$. These findings indicate that the common practice of setting the parameter $m$ of the constrained DTW-MEAN problem to the length $n$ of the sample time series results in a low structural error on median. Due to the low median slopes on $D_2$, solving the constrained DTW-MEAN problem instead of the unconstrained DTW-MEAN problem can in principle give good approximations for values $m > m_{cm}$. The average and maximum values for the slopes and error percentages, however, show that there are outliers that result in large structural error between 37 and 120%.

6.2.3 Shape

One problem that is often associated with dynamic time warping is shape averaging. Niennattrakul and Ratanamahatana (2009) proposed a shape averaging algorithm that
Fig. 8  
(a) Sketches the main characteristics of all observed shapes of \( F^*_m \) on \( D_1 \) and \( D_2 \) together with their percentage frequency. Sketches exaggerate bumps and slopes to better highlight the main features. 
(b) A typical example of the most common shape of \( F^*_m \) for a sample of two random walks of length \( n = 40 \). The function \( F^*_m \) strictly decreases on \( D_1 \) until it reaches its global minimum at \( m_{\text{cm}} = 32 \) as highlighted by the red circle. Then the function \( F^*_m \) strictly increases on \( D_2 \). The best linear fit of \( F^*_m \) on \( D_2 \) has a very small slope of approximately \( 10^{-5} \) (Color figure online)

Table 3  Slopes \( a \) of best linear fits of the second part of \( F^*_m \) and error percentages \( E \) of the structural error

| \( n \) | \( a_{\text{med}} \) | \( a_{\text{avg}} \) | \( a_{\text{std}} \) | \( a_{\text{max}} \) | \( E_{\text{med}} \) | \( E_{\text{avg}} \) | \( E_{\text{std}} \) | \( E_{\text{max}} \) |
|-------|----------------|-------------|-------------|-------------|-------------|-------------|-------------|-------------|
| 5     | 0.0034         | 0.12        | 0.36        | 3.78        | 0.423       | 4.43        | 9.59        | 67.40       |
| 10    | 0.0008         | 0.05        | 0.25        | 3.37        | 0.057       | 2.13        | 8.50        | 119.42      |
| 15    | 0.0005         | 0.05        | 0.24        | 2.80        | 0.028       | 1.49        | 7.30        | 92.65       |
| 20    | 0.0002         | 0.03        | 0.14        | 1.83        | 0.012       | 0.69        | 3.36        | 40.22       |
| 25    | 0.0002         | 0.02        | 0.14        | 3.15        | 0.008       | 0.47        | 2.77        | 37.70       |
| 30    | 0.0001         | 0.01        | 0.10        | 1.63        | 0.004       | 0.37        | 2.42        | 40.37       |
| 35    | 0.0001         | 0.01        | 0.06        | 1.35        | 0.003       | 0.20        | 2.22        | 52.44       |
| 40    | 0.0001         | 0.01        | 0.12        | 2.16        | 0.002       | 0.28        | 2.24        | 51.20       |

Subscripts \( \text{med}, \text{avg}, \text{std}, \text{max} \) refer to the median, average, standard deviation, and maximum, respectively, of 1000 pairs of random walks of length \( n \)
uses dynamic time warping, but differs from minimizing the Fréchet function. Sun et al. (2017) claimed that their approach of minimizing the Fréchet function preserves shape characteristics. Cuturi and Blondel (2017) criticized that solutions found by DTW barycenter averaging (DBA) (Hautamaki et al. 2008; Petitjean et al. 2011) may have a shape that is not representative for the sample time series.

By construction, dynamic time warping is invariant to distortions in time, but not in scale (amplitude). For this reason, similar shapes of two time series may not be warped onto each other if their amplitudes differ (Keogh and Pazzani 2001; Zhao and Itti 2018). In this section, we demonstrate that this effect transfers to means in dtw-spaces, implying that a mean does not necessarily preserve shapes of the sample time series. Note that this is a non-judgemental observation; the question whether this property is undesirable or not depends on the application at hand.

**Setup** We selected two sample time series $x_1$ and $x_2$ from the UCR data set ECG200. Both time series have zero mean and standard deviation one. For $x \in [x_1, x_2]$ we applied Algorithm 1 to compute a mean of $x_1$ and $x_2$ and plotted all time series. The last shift of $x_1$ yields equal minimum values of $x_1$ and $x_2$.

**Results and discussion** Figures 9, 10 and 11 show a mean of the two sample time series $x_1$ and $x_2$. In Fig. 9a the mean has a small kink near time index 30. This kink has also been observed in the solution found by DBA in a similar experiment (Cuturi and Blondel 2017). Cuturi and Blondel (2017) criticized that this kink is not present in any of the sample time series. We now can conclude that presence of kinks is a peculiarity of means in dtw-spaces, rather than a property of the heuristics. Figure 9b illustrates the warpings near the kink. The kink can be explained as follows: Both sample time series have broad valleys but on a different scale. In this case, the minimum point of the upper time series $x_1$ is warped through mean elements onto multiple points of the lower time series $x_2$, which results in a kink in the mean.

To strengthen this effect, we used the shifted time series $x_1 = x_1 + 1$ in Fig. 10. As expected, the kink becomes larger, because more points of $x_2$ are warped onto
a single point of \( x^{(1)} \) through mean elements (Fig. 10b). Moreover, the length of the mean reduces significantly.

To weaken this effect, we equalized the minima of the sample time series in Fig. 11. Now the kink disappears, because the warping of the valleys of \( x^{(1)} \) and \( x^{(2)} \) is more balanced (Fig. 11b). Note however that the valley of the mean is more similar to the valley of \( x^{(1)} \) than to the valley of \( x^{(2)} \).

These experiments suggest that a mean in dtw-space is not necessarily a suitable representative of the sample time series in terms of shape, particularly if the shapes appear on different scale (amplitude) in the sample time series.

### 6.2.4 Summary

The main findings of our first series of experiments are:

1. A mean of a sample is usually condensed, unique and shorter than the sample time series.
2. Choosing a mean length that is larger than the optimal value induces only a small structural error in most cases.
3. A mean might not necessarily preserve shapes of the sample time series (although further experiments should be performed to confirm this observation).

The first two points suggest that critical problems in applications caused by non-unique means or by the common practice to constrain the length of a mean to the length of the sample time series are exceptional. The third point suggests that more representative shapes of a mean may require different objective functions to be minimized or different distance measures [as for example derivative dtw (Keogh and Pazzani 2001), shapeDTW (Zhao and Itti 2018) or soft-dtw (Cuturi and Blondel 2017)].

6.3 Performance of heuristics

The goal of this series of experiments is to assess the performance of state-of-the-art heuristics in terms of minimizing the Fréchet variation. This allows to obtain an estimate of how well existing heuristics approximate an exact mean.

6.3.1 Experimental setup

**Algorithms:** Table 4 lists the mean-algorithms considered in this experiment.

Section 3.2 discusses MAL. Here, we consider MAL for means of only two time series, because this method is not clearly defined in the literature for \( k > 2 \).

The DBA algorithm (Hautamaki et al. 2008; Petitjean et al. 2011) is a majorize-minimize procedure (Schultz and Jain 2018). The majorization-step computes optimal warpings between the current solution and the sample time series. The minimization-step averages the warped sample time series with respect to the necessary conditions of optimality (Schultz and Jain 2018). The DBA algorithm converges after a finite number of iterations to a local minimum (Schultz and Jain 2018).

The subgradient methods BSG and SSG are based on the observation that the Fréchet function is locally Lipschitz continuous and therefore differentiable almost everywhere (Schultz and Jain 2018). For locally Lipschitz functions, subgradient methods in the sense of Clarke (1990) from nonsmooth optimization can be applied. The BSG and SSG algorithms refer to the vanilla batch and stochastic subgradient methods for locally Lipschitz functions.

| Algorithm | Acronym |
|-----------|---------|
| Exact dynamic programming (Algorithm 1) | EDP |
| Multiple alignment (Hautamaki et al. 2008; Petitjean and Gançarski 2012) | MAL |
| DTW barycenter averaging (Hautamaki et al. 2008; Petitjean et al. 2011) | DBA |
| Soft-dtw (Cuturi and Blondel 2017) | SDTW |
| Batch subgradient (Cuturi and Blondel 2017; Schultz and Jain 2018) | BSG |
| Stochastic subgradient (Schultz and Jain 2018) | SSG |
The SDTW algorithm (Cuturi and Blondel 2017) is based on a smoothed version of the dtw-distance by unifying the standard dtw-distance with a global alignment kernel proposed by Cuturi et al. (2007). The resulting soft-dtw distance is differentiable if its smoothing parameter $\gamma$ is positive and recovers the original dtw-distance for $\gamma = 0$. Hence, for $\gamma > 0$ the Fréchet function with respect to soft-dtw is also differentiable (note however that this is a different objective, which is why SDTW should not be expected to yield optimal means). The SDTW method applies L-BFGS to minimize the Fréchet function.

We implemented EDP and MAL (for two time series) in Java. For SDTW and BSG we used the Python implementation provided by Blondel (2017). For DBA and SSG we used the implementation in Java (Schultz and Jain 2016).

**Setup** All six algorithms were applied to 27,000 samples of type $S_{\text{ucr}}$ and to 10,000 samples of type $S_{\text{rw}}$. For the 5000 samples of type $S_{\text{rw}}^k$, all mean algorithms but MAL (as explained in the previous paragraph) were applied.

EDP and MAL required no parameter optimization. For the other algorithms, the following settings were used: We optimized the algorithms on every sample using different parameter configurations and reported the best result in terms of minimizing the Fréchet function. Every configuration was composed of an initialization method and an optional parameter selection. As initialization we used (1) the arithmetic mean of the sample, (2) a random time series from the sample, and (2) a random time series drawn from a normal distribution with zero mean and standard deviation one. DBA and BSG required no additional parameter selection. For SDTW, we selected the best smoothing parameter $\gamma \in \{0.001, 0.01, 0.1, 1\}$ and for SSG the best initial step size $\eta_0 \in \{0.25, 0.2, 0.15, 0.1\}$. The step size of SSG was linearly decreased from $\eta_0$ to $\eta_0/10$ within the first 100 epochs and then remained constant for the remaining 100 epochs (one epoch is a full pass through the sample). Thus, for DBA and BSG, we picked the best result from the three initialization methods for each sample, and for SDTW and SSG we picked the best result from twelve parameter configurations (three initializations $\times$ four parameter values). The length of the mean was set beforehand to the length of the sample time series (time series of a sample always have identical length). For every sample and every parameter configuration, all algorithms terminated after 200 epochs at the latest. The tolerance parameter of SDTW and BSG was set to $\varepsilon = 10^{-6}$.

**Performance metric** We consider error percentages to assess the solution quality. The error percentage of algorithm $A$ for sample $S$ is defined by

$$E = 100 \cdot \frac{(F_A - F_*)}{F_*},$$

where $F_A$ is the solution obtained by algorithm $A$ and $F_* = \min_{z \in T} F(z)$ is the optimal solution obtained by EDP.
Fig. 12 Results for $S_{ucr}$-samples. (left) Performance profiles of all heuristics (a larger area under the curve indicates better performance). The performance profile of the exact algorithm EDP is the constant line $P = 1$ and therefore not highlighted. The other performance profiles are truncated at 200% error for the sake of presentation. A point $(E_A, P_A)$ on the curve of an algorithm $A$ states that solutions obtained by $A$ deviate by at most $E_A$ percent from the optimal solution with probability $P_A$ (estimated over 27,000 different samples). (right) Average error percentage (avg), standard deviation (std), and maximum error percentage (max) of the heuristics. The last column shows how often an optimal solution was found by the heuristic.

Fig. 13 Results for $S_{rw}$-samples. (left) Performance profiles of all heuristics. (right) Average error percentage (avg), standard deviation (std), and maximum error percentage (max) of the heuristics. The last column shows how often an optimal solution was found by the heuristic.

6.3.2 Results and discussion

Figures 12 and 13 depict the performance profiles of the algorithms and quantitative summaries of their error percentages for samples of type $S_{ucr}$ and $S_{rw}$, respectively. The performance profiles exhibit the same pattern in general but differ by some shifts in the curves which is due to different data domains (Tables 5, 6, 7 in “Appendix B” present the results in more detail for the respective data sets and values of $k$).

Performance of multiple alignment (MAL) Section 3.2 refutes the claim that DTW-MEAN can be solved by a multiple alignment approach. The remaining question is to which extent MAL fails to solve DTW-MEAN.

---

2 “Appendix A” describes performance profiles in more detail.
Figures 12 and 13 show that MAL exactly solved DTW-MEAN in only three out of 27,000 $S_{ucr}$-samples and another three out of 10,000 $S_{rw}$-samples. The average error percentages of MAL on $S_{ucr}$ and $S_{rw}$ are 68.3 and 210.7, respectively. The error percentage of MAL is larger than 50% for more than 40% of all $S_{ucr}$-samples and for roughly 90% of all $S_{rw}$-samples. These observations suggest that MAL is far from being optimal or competitive. This may explain why algorithms based on pairwise mean computation via MAL such as NLAAF (Gupta et al. 1996) and PSA (Niennattrakul and Ratanamahatana 2009) are not competitive (Petitjean et al. 2011; Soheily-Khah et al. 2015).

Performance of state-of-the-art heuristics We discuss the performance of SDTW, BSG, DBA, and SSG. The best and most robust method is the stochastic subgradient method (SSG) with 15.1% and 12.8% average error percentage on $S_{ucr}$- and $S_{rw}$-samples, respectively. Standard deviation and maximum error percentages of SSG are lowest among all other heuristics. Nevertheless, the solution quality of all heuristics is rather poor leaving much space for improvements.

Performance of progressive alignment (PSA) Progressive alignment methods combine pairwise averages beginning with the two most similar time series. In each step, two time series are averaged and replaced by their weighted average. The weights correspond to the number of sample time series involved in creating the time series. The currently best performing progressive alignment method for time series averaging is called prioritized shape averaging (PSA) (Niennattrakul and Ratanamahatana 2009). This method applies a variant of MAL for pairwise averaging. Empirical results suggest that PSA is not competitive to DBA (Petitjean et al. 2011; Soheily-Khah et al. 2015). We evaluate the performance of a modified variant of PSA using EDP for pairwise averaging.

As test data, we used samples of type $S_{ucr}^k$: for every UCR data set and every sample size $k \in \{5, 10, 15, 20\}$, we randomly selected 100 samples of $k$ time series giving a total of 10,800 samples. We applied the modified PSA algorithm and the algorithms SDTW, BSG, DBA, and SSG on all 10,800 samples (the setup is the same as described in Sect. 6.3.1).

Figure 14 summarizes the results. The performance profiles exhibit similar patterns for $S_{ucr}^k$-samples as those obtained for $S_{ucr}$-samples (Fig. 12). PSA is the worst performing heuristic. Figure 14b shows that the average error percentage of PSA increases with increasing sample size $k$, whereas the average error percentage of all other heuristics exhibit the opposite trend. These results show that the weak performance of progressive alignment methods is not explained by suboptimal pairwise mean computation.

Error decomposition The previous results revealed a poor solution quality of state-of-the-art heuristics. In Sect. 6.2.2, we have seen that choosing a fixed mean-length $m$ introduces a certain structural error. The error $\varepsilon$ of an algorithm $A$ can thus be written as

$$\varepsilon = F_A - F_\ast = \left( F_A - F_\ast^m \right) + \left( F_\ast^m - F_\ast \right),$$

where $$\varepsilon_m$$ approximates the error and $$\varepsilon_m$$ approximates the structural error.
where \( F_A \) is the value returned by algorithm \( A \), \( F_* = \min_{z \in T} F(z) \) is the Fréchet variation, and \( F^*_m = \min_{z \in T_m} F(z) \) is the constrained Fréchet variation. The approximation error \( F_A - F^*_m \) represents the inability of algorithm \( A \) to solve the constrained DTW-MEAN problem.

We conducted some experiments to assess the contribution of the approximation and the structural error to the total error of the considered heuristics. For every length \( n \in \{5, 10, 15, \ldots, 40\} \), we generated 1000 samples of pairs of random walks of length \( n \) giving a total of 8000 samples. We applied SDTW, BSG, DBA, and SSG on all samples using the same setting as described in Sect. 6.3.1.

Figure 15 depicts the results of the best performing heuristic (SSG). The other three heuristics exhibited the same behavior with identical structural error but different approximation errors. We made the following observations:

1. The total error is dominated by the approximation error (for larger \( n \)).
2. Both total and approximation error increase with increasing length \( n \).
3. The structural error is independent of the length \( n \).
Results on samples of type $S_{\text{rw}}$ consisting of $k = 2$ random walks of length $n$. Average error percentage (avg), standard deviation (std), maximum error percentage (max) and number of exact solutions are plotted as a function of the length $n$.

4. In exceptional cases the structural error can be large.

These observations indicate that the total error is mainly caused by the approximation error. In contrast, the structural error introduced by fixing the mean length becomes negligible for longer sample time series and is large only in very few cases (as concluded in Sect. 6.2.4). The results suggest that devising heuristics should mainly focus on improving the approximation error.

**Effect of sample time series length $n$** We investigated how the performance of SDTW, BSG, DBA, and SSG is related to the length $n$ of the sample time series for random walk samples of type $S_{\text{rw}}$.

Figure 16 summarizes the results. The trend is that the performance of all heuristics with respect to solution quality (avg) and robustness (std, max) deteriorates with increasing length $n$. Since we observed before that the structural error is independent of $n$ and likely to be negligible, we conclude that the heuristics perform worse in solving the constrained DTW-Mean problem for larger $n$. No state-of-the-art method found an optimal solution for length $n \geq 20$.

**Effect of sample size $k$** We analyzed how the performance of SDTW, BSG, DBA, and SSG depends on the sample size $k$ for samples of type $S_{\text{rw}}^k$.

Figure 17 shows the results. We observed no clear trend with respect to average error-percentage and a decline of the number of exact solutions found by all heuristics.
In contrast, robustness (that is, standard deviation and maximum error percentage) of all four heuristics substantially improved with increasing $k$.

6.3.3 Summary

The main findings of our second series of experiments concerning the quality of heuristics for solving DTW-MEAN are as follows.

1. State-of-the-art heuristics do not perform well on average (on small samples) due to a large approximation error of around 15–30%.
2. MAL cannot compete with state-of-the-art heuristics (for $k = 2$).
3. PSA using exact weighted means cannot compete with state-of-the-art heuristics.

Concerning running times, our exact dynamic program is much slower than state-of-the-art heuristics (for example, our implementation required 66 seconds on average to compute a mean of two time series of length 100, which is slower by a factor of $10^5$ compared to DBA). Hence, for real-world applications this approach is not feasible.

7 Conclusion

We developed an exact exponential-time algorithm for WEIGHTED DTW-MEAN and conducted extensive experiments (on small samples containing up to six time series...
of different lengths up to 150) to investigate characteristics of an exact mean and to benchmark existing state-of-the-art heuristics in terms of their solution quality. On the positive side, we found that a mean is often unique and that fixing the length of a mean in advance does usually not affect the solution quality much. On the negative side, we showed that basically all heuristics perform poorly in the worst case. These findings urge for devising better algorithms to solve DTW-MEAN. Our empirical observations (e.g., concerning the typical mean length) might prove useful for this. As a side result, we also developed a polynomial-time algorithm for binary time series.

We conclude with some challenges for future research. From an algorithmic point of view it is interesting to investigate whether one can extend the polynomial-time solvability of BINARY WEIGHTED DTW-MEAN to larger “alphabet” sizes; already the case of alphabet size three is open. Another open question is whether WEIGHTED DTW-MEAN is polynomial-time solvable for time series of constant lengths (maybe it is even fixed-parameter tractable with respect to the maximum length). Finally, we wonder whether there are other practically relevant restrictions of DTW-MEAN that make the problem more tractable, for example, fixing the length of a mean. Another example, also motivated from a practical point of view, is to compute means with a given size of time windows [also known as the Sakoe–Chiba band (Sakoe and Chiba 1978)]. On a high level, a time window constrains the warping path to only select tuples within a specified range.

On an applied side, our experimental results strongly motivate the search for further improved, more “robust” heuristics. Also, it is not clear whether an exact mean actually performs better in applications such as clustering (e.g., \(k\)-means) or classification. However, empirical findings (Cuturi et al. 2007; Petitjean et al. 2011) suggest that better approximations of a mean result in lower \(k\)-means loss under dtw-distance. Answering this question is challenging since computing exact solutions quickly becomes intractable for growing data size.

Acknowledgements This work was supported by the Deutsche Forschungsgemeinschaft under Grants JA 2109/4-1 and NI 369/13-2, and by a Feodor Lynen return fellowship of the Alexander von Humboldt Foundation. The work on the theoretical part of this paper started at the research retreat of the Algorithmics and Computational Complexity group, TU Berlin, held at Boiensdorf, Baltic Sea, April 2017, with MB, TF, VF, and RN participating. We also thank the authors of the UCR Time Series Classification Archive for providing the data sets which we used in our experiments.

A Performance profiles

To compare the performance of the mean algorithms, we used a slight variation of the performance profiles proposed by Dolan and Moré (2002). A performance profile is a cumulative distribution function for a performance metric. Here, the chosen performance metric is the error percentage from the exact solution.

To define a performance profile, we assume that \(\mathcal{A}\) is a set of mean algorithms and \(\mathcal{S}\) is a set of samples each of which consists of \(k\) time series. For each sample \(\mathcal{X} \in \mathcal{X}\) and each mean algorithm \(A \in \mathcal{A}\), we define \(E_{A,S}\) as the error percentage obtained by applying algorithm \(A\) on sample \(S\). The performance profile of algorithm \(A\) over all samples \(S \in \mathcal{S}\) is the empirical cumulative distribution function defined by
\[ P_A(\tau) = \frac{1}{|S|} \left| \{ S \in S : E_{A,S} \leq \tau \} \right| \]

for all \( \tau \geq 0 \). Thus, \( P_A(\tau) \) is the estimated probability that the error percentage of algorithm \( A \) is at most \( \tau \). The value \( P_A(0) \) is the estimated probability that algorithm \( A \) finds an exact solution.

## B Detailed results

See Tables 5, 6 and 7.

### Table 5  Average error percentage of the five heuristics on samples \( S_{ucr} \) of size \( k = 2 \) grouped by UCR data set

| UCR data set                  | MAL  | SDTW | BSG  | DBA  | SSG  |
|-------------------------------|------|------|------|------|------|
| ItalyPowerDemand              | 26.1 | 13.8 | 17.6 | 19.2 | 10.5 |
| Synthetic_control             | 54.7 | 22.1 | 26.5 | 29.4 | 17.9 |
| SonyAIBORobotSurface          | 33.6 | 20.6 | 26.0 | 28.9 | 17.2 |
| SonyAIBORobotSurfaceII        | 28.4 | 20.0 | 23.8 | 26.7 | 17.8 |
| ProximalPhalanxTW             | 35.2 | 11.8 | 15.3 | 17.1 | 10.6 |
| ProximalPhalanxOutlineCorrect | 35.1 | 13.4 | 16.9 | 18.4 | 11.9 |
| ProximalPhalanxOutlineAgeGroup| 36.9 | 11.7 | 15.3 | 17.4 | 10.8 |
| PhalangesOutlinesCorrect      | 49.7 | 15.2 | 19.5 | 22.3 | 13.2 |
| MiddlePhalanxTW               | 36.8 | 12.5 | 14.8 | 17.3 | 10.0 |
| MiddlePhalanxOutlineCorrect   | 38.7 | 13.3 | 16.2 | 18.4 | 10.8 |
| MiddlePhalanxOutlineAgeGroup  | 37.0 | 12.8 | 15.1 | 17.6 | 9.9  |
| DistalPhalanxTW               | 41.9 | 12.0 | 15.3 | 18.6 | 10.5 |
| DistalPhalanxOutlineCorrect   | 48.4 | 13.4 | 17.4 | 21.2 | 11.6 |
| DistalPhalanxOutlineAgeGroup  | 43.6 | 12.4 | 15.5 | 18.6 | 10.6 |
| TwoLeadECG                    | 44.4 | 12.0 | 15.3 | 17.7 | 9.6  |
| MoteStrain                    | 79.7 | 17.5 | 23.3 | 29.2 | 12.6 |
| ECG200                        | 51.4 | 20.5 | 23.5 | 26.5 | 14.2 |
| MedicalImages                 | 84.6 | 16.8 | 21.0 | 27.6 | 10.6 |
| Two_Patterns                  | 184.0| 73.5 | 82.8 | 92.8 | 56.4 |
| SwedishLeaf                   | 56.2 | 18.0 | 23.2 | 30.6 | 13.4 |
| CBF                           | 28.5 | 31.2 | 33.8 | 36.9 | 26.4 |
| FacesUCR                      | 38.4 | 22.5 | 27.6 | 30.2 | 18.0 |
| FaceAll                       | 37.7 | 22.6 | 27.6 | 30.2 | 18.3 |
| ECGFiveDays                   | 41.7 | 11.0 | 13.0 | 15.9 | 8.5  |
| ECG5000                       | 77.8 | 15.3 | 18.0 | 22.4 | 9.9  |
| Plane                         | 83.1 | 13.6 | 18.6 | 30.0 | 9.7  |
Table 5 continued

| UCR data set | MAL | SDTW | BSG | DBA | SSG |
|--------------|-----|------|-----|-----|-----|
| Gun_Point    | 489.8 | 36.2 | 52.0 | 77.0 | 27.4 |
| Total        | 68.3 | 19.1 | 23.5 | 28.1 | 15.1 |

Averages were taken over 1000 samples randomly drawn from each data set.

Table 6 Average error percentage of the five heuristics on $S_{rw}$-samples of size $k = 2$ grouped by length $n$ of random walks

| $n$ | MAL | SDTW | BSG | DBA | SSG |
|-----|-----|------|-----|-----|-----|
| 10  | 64.3 | 9.6 | 13.2 | 15.8 | 8.2 |
| 20  | 107.4 | 14.2 | 19.4 | 23.4 | 9.2 |
| 30  | 152.1 | 18.0 | 23.7 | 28.0 | 10.2 |
| 40  | 179.1 | 19.9 | 24.7 | 32.1 | 11.4 |
| 50  | 206.5 | 22.4 | 28.8 | 37.2 | 12.5 |
| 60  | 231.2 | 24.7 | 29.9 | 38.7 | 13.9 |
| 70  | 253.2 | 26.4 | 32.2 | 40.5 | 14.6 |
| 80  | 269.7 | 28.3 | 32.6 | 43.2 | 15.0 |
| 90  | 303.9 | 29.4 | 34.5 | 47.7 | 15.8 |
| 100 | 339.8 | 31.9 | 34.4 | 53.4 | 17.1 |
| Total | 210.7 | 22.5 | 27.4 | 36.0 | 12.8 |

Averages were taken over 1000 samples for each length $n$.

Table 7 Average error percentage of the five heuristics on $S_{rw}^k$-samples of varying sample size $k$

| $k$ | Avg | Std | Max | Eq |
|-----|-----|-----|-----|----|
| $k = 2$ | | | | |
| SDTW | 7.0 | 10.8 | 72.8 | 68.0 |
| BSG  | 10.2 | 13.7 | 82.3 | 80.0 |
| DBA  | 11.4 | 15.0 | 120.8 | 76.0 |
| SSG  | 7.9 | 12.5 | 120.8 | 56.0 |
| $k = 3$ | | | | |
| SDTW | 7.7 | 9.5 | 63.7 | 22.0 |
| BSG  | 9.8 | 11.0 | 63.7 | 25.0 |
| DBA  | 10.7 | 11.3 | 67.9 | 20.0 |
| SSG  | 7.2 | 9.9 | 72.9 | 14.0 |
| $k = 4$ | | | | |
| SDTW | 8.9 | 9.1 | 51.8 | 9.0 |
| BSG  | 10.4 | 9.8 | 52.8 | 7.0 |
Exact mean computation in dynamic time warping spaces

|            | Avg | Std | Max | Eq |
|------------|-----|-----|-----|----|
| DBA        | 11.4| 10.6| 62.0| 4.0|
| SSG        | 8.4 | 9.1 | 57.0| 3.0|

$k = 5$

|            | Avg | Std | Max | Eq |
|------------|-----|-----|-----|----|
| SDTW       | 9.8 | 8.9 | 51.6| 3.0|
| BSG        | 11.2| 9.6 | 51.6| 3.0|
| DBA        | 12.0| 9.9 | 57.5| 3.0|
| SSG        | 9.3 | 8.8 | 48.1| 2.0|

$k = 6$

|            | Avg | Std | Max | Eq |
|------------|-----|-----|-----|----|
| SDTW       | 9.8 | 7.5 | 46.7| 2.0|
| BSG        | 11.0| 8.2 | 46.7| 2.0|
| DBA        | 11.5| 8.6 | 46.7| 1.0|
| SSG        | 9.3 | 7.8 | 46.7| 0.0|

Length of random walks was $n = 6$. Averages were taken over 1000 samples for each sample size $k$

References

Aach J, Church GM (2001) Aligning gene expression time series with time warping algorithms. Bioinformatics 17(6):495–508

Abdulla WH, Chow D, Sin G (2003) Cross-words reference template for DTW-based speech recognition systems. In: Proceedings of the IEEE conference on convergent technologies for the Asia-Pacific region (TENCON 2003), vol 4. IEEE, pp 1576–1579

Aghabozorgi S, Shirorkhoshidi AS, Wah TY (2015) Time-series clustering—a decade review. Inf Syst 53:16–38

Alon J, Athitos V, Yuan Q, Sclaroff S (2009) A unified framework for gesture recognition and spatiotemporal gesture segmentation. IEEE Trans Pattern Anal Mach Intell 31(9):1685–1699

Blondel M (2017) Python implementation of soft-DTW. https://github.com/mblondel/soft-dtw

Bonizzoni P, Della Vedova G (2001) The complexity of multiple sequence alignment with SP-score that is a metric. Theor Comput Sci 259(1–2):63–79

Bringmann K, Künnemann M (2015) Quadratic conditional lower bounds for string problems and dynamic time warping. In: Proceedings of the 56th annual IEEE symposium on foundations of computer science (FOCS ’15). IEEE, pp 79–97

Bulteau L, Froese V, Niedermeier R (2018) Tight hardness results for consensus problems on circular strings and time series. CoRR arXiv:1804.02854

Chen Y, Keogh E, Hu B, Begum N, Bagnall A, Mueen A, Batista G (2015) The UCR time series classification archive. www.cs.ucr.edu/~eamonn/time_series_data/

Clarke FH (1990) Optimization and nonsmooth analysis. SIAM, Philadelphia

Cuturi M, Blondel M (2017) Soft-DTW: a differentiable loss function for time-series. In: Proceedings of the 34th international conference on machine learning (ICML ’17). PMLR, proceedings of machine learning research, vol 70, pp 894–903

Cuturi M, Vert JP, Birkenes O, Matsui T (2007) A kernel for time series based on global alignments. In: Acoustics, speech and signal processing, 2007. ICASSP 2007. IEEE international conference on, vol 2. IEEE, pp 2–413

Dolan E, Moré J (2002) Benchmarking optimization software with performance profiles. Math Program 91(2):201–213

Fahiman F, Bezdek JC, Erfani SM, Leckie C, Palaniswami M (2017) Fuzzy c-Shape: a new algorithm for clustering finite time series waveforms. In: Proceeding of the 2017 IEEE international conference on fuzzy systems (FUZZ-IEEE). IEEE, pp 1–8
Fréchet M (1948) Les éléments aléatoires de nature quelconque dans un espace distancié. Ann Inst Henri Poincaré 20:215–310

Fu T (2011) A review on time series data mining. Eng Appl Artif Intell 24(1):164–181

Gold O, Sharir M (2018) Dynamic time warping and geometric edit distance: breaking the quadratic barrier. ACM Trans Algorithms 14(4):50:1–50:17

Gupta L, Molfese DL, Tammana R, Simos PG (1996) Nonlinear alignment and averaging for estimating the evoked potential. IEEE Trans Biomed Eng 43(4):348–356

Gusfield D (1997) Algorithms on strings, trees and sequences. Cambridge University Press, Cambridge

Hautamaki V, Nykanen P, Franti P (2008) Time-series clustering by approximate prototypes. In: Proceedings of the 19th international conference on pattern recognition (ICPR ’08). IEEE, pp 1–4

Huang B, Kinsner W (2002) ECG frame classification using dynamic time warping. In: Canadian conference on electrical and computer engineering (CCECE), vol 2. IEEE, pp 1105–1110

Jain B, Schultz D (2017) Optimal warping paths are unique for almost every pair of time series. CoRR arXiv:1705.05681

Jain B, Schultz D (2018) On the existence of a sample mean in dynamic warping spaces. CoRR arXiv:1610.04460v3

Keogh EJ, Pazzani MJ (2001) Derivative dynamic time warping. In: Proceedings of the 2001 SIAM international conference on data mining. SIAM, Philadelphia, pp 1–11

Lummis R (1973) Speaker verification by computer using speech intensity for temporal registration. IEEE Trans Audio Electroacoust 21(2):80–89

Marteau PF (2018) Times series averaging and denoising from a probabilistic perspective on time-elastic kernels. Int J Appl Math Comput Sci. arXiv:1611.09194

Moon TK (1996) The expectation-maximization algorithm. IEEE Signal Process Mag 13(6):47–60

Morel M, Achard C, Kulpa R, Dubuisson S (2018) Time-series averaging using constrained dynamic time warping with tolerance. Int J Pattern Recogn 74:77–89

Myers CS, Rabiner LR (1981) A comparative study of several dynamic time-warping algorithms for connected-word recognition. Bell Syst Tech J 60(7):1389–1409

Nicolas F, Rivals E (2005) Hardness results for the center and median string problems under the weighted and unweighted edit distances. J Discrete Algorithms 3(2):390–415

Niennattrakul V, Ratanamahatana CA (2007) Inaccuracies of shape averaging method using dynamic time warping for time series data. In: International conference on computational science. Springer, Berlin, pp 513–520

Niennattrakul V, Ratanamahatana CA (2009) Shape averaging under time warping. In: Shi Y, van Albada GD, Dongarra J, Sloot P (eds) Proceedings of the 6th international conference on electrical engineering/electronics, computer, telecommunications and information technology (ECTI-CON ’09), vol 02, pp 626–629

Oates T, Firoiu L, Cohen PR (1999) Clustering time series with hidden Markov models and dynamic time warping. In: Proceedings of the IJCAI’99 workshop on neural, symbolic and reinforcement learning methods for sequence learning, pp 17–21

Paparrizos J, Gravano L (2017) Fast and accurate time-series clustering. ACM Trans Database Syst 42(2):8:1–8:49

Petitjean F, Gančarski P (2012) Summarizing a set of time series by averaging: from Steiner sequence to compact multiple alignment. Theor Comput Sci 414(1):76–91

Petitjean F, Ketterlin A, Gančarski P (2011) A global averaging method for dynamic time warping, with applications to clustering. Pattern Recogn 44(3):678–693

Petitjean F, Forestier G, Webb GI, Nicholson AE, Chen Y, Keogh E (2014) Dynamic time warping averaging of time series allows faster and more accurate classification. In: Proceeding of the 2014 IEEE international conference on data mining (ICDM’14). IEEE, pp 470–479

Petitjean F, Forestier G, Webb GI, Nicholson AE, Chen Y, Keogh E (2016) Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. Knowl Inf Syst 47(1):1–26

Rabiner L, Wilpon J (1979) Considerations in applying clustering techniques to speaker-independent word recognition. J Acoust Soc Am 66(3):663–673

Reyes M, Dominguez G, Escalera S (2011) Feature weighting in dynamic timewarping for gesture recognition in depth data. In: Proceeding of the 2011 IEEE international conference on computer vision workshops (ICCV Workshops). IEEE, pp 1182–1188
Sakoe H, Chiba S (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE Trans Acoust Speech Signal Process 26(1):43–49
Sankoff D, Kruskal J (1999) Time warps, string edit and macromolecules: the theory and practice of sequence comparison. CSLI Publications, Stanford
Schultz D, Jain BJ (2016) Sample mean algorithms for averaging in dynamic time warping spaces. https://doi.org/10.5281/zenodo.216233
Schultz D, Jain B (2018) Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces. Pattern Recogn 74(Supplement C):340–358
Soheily-Khah S, Douzal-Chouakria A, Gaussier E (2015) Progressive and iterative approaches for time series averaging. In: Proceedings of the 1st international conference on advanced analytics and learning on temporal data, vol 1425, pp 111–117
Soheily-Khah S, Douzal-Chouakria A, Gaussier E (2016) Generalized $k$-means-based clustering for temporal data under weighted and kernel time warp. Pattern Recogn Lett 75:63–69
Sun T, Liu H, Yu H, Chen CP (2017) Degree-pruning dynamic programming approaches to central time series minimizing dynamic time warping distance. IEEE Trans Cybern 47(7):1719–1729
Zhao J, Itti L (2018) shapeDTW. Pattern Recogn 74(C):171–184

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Markus Brill1 · Till Fluschnik2 · Vincent Froese2 · Brijnesh Jain3 · Rolf Niedermeier2 · David Schultz2

Markus Brill
brill@tu-berlin.de
Till Fluschnik
till.fluschnik@tu-berlin.de
Brijnesh Jain
brijnesh.jain@dai-labor.de
Rolf Niedermeier
rolf.niedermeier@tu-berlin.de
David Schultz
david.schultz@dai-labor.de

1 Efficient Algorithms, TU Berlin, Berlin, Germany
2 Algorithmics and Computational Complexity, TU Berlin, Berlin, Germany
3 Distributed Artificial Intelligence Laboratory, TU Berlin, Berlin, Germany