Nonsmooth Analysis and Subgradient Methods for Averaging in Dynamic Time Warping Spaces

David Schultz and Brijnesh Jain
Technische Universität Berlin, Germany

Abstract. Time series averaging in dynamic time warping (DTW) spaces has been successfully applied to improve pattern recognition systems. This article proposes and analyzes subgradient methods for the problem of finding a sample mean in DTW spaces. The class of subgradient methods generalizes existing sample mean algorithms such as DTW Barycenter Averaging (DBA). We show that DBA is a majorize-minimize algorithm that converges to necessary conditions of optimality after finitely many iterations. Empirical results show that for increasing sample sizes the proposed stochastic subgradient (SSG) algorithm is more stable and finds better solutions in shorter time than the DBA algorithm on average. Therefore, SSG is useful in online settings and for non-small sample sizes. The theoretical and empirical results open new paths for devising sample mean algorithms: nonsmooth optimization methods and modified variants of pairwise averaging methods.

Contents

1. Introduction

Inspired by the sample mean in Euclidean spaces, the goal of time series averaging is to construct a time series that is located in the center of a given sample of time series. One common technique to average time series is based on first aligning the time series and then synthesizing the aligned time series to an average. In doing so, the alignment of time series is typically performed with respect to the dynamic time warping (DTW) distance. Several variations of this approach have been successfully applied to improve nearest neighbor classifiers and to formulate centroid-based clustering algorithms in DTW spaces [1, 13, 22, 27, 28, 33].

Among the different variations, one research direction poses time series averaging as an optimization problem [13, 24, 33]: Suppose that $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right)$ is a sample of N time series $x^{(i)}$. Then a (sample) mean in DTW spaces is any time series that minimizes the Fréchet function [10]

$$F(x) = \frac{1}{N} \sum_{k=1}^{N} \text{dtw}^2\left(x, x^{(k)}\right),$$

where dtw is the DTW distance. A polynomial-time algorithm for finding a global minimum of the non-differentiable, non-convex Fréchet function is unknown. Therefore, an ongoing research problem is to devise improved approximation algorithms that efficiently find acceptable sub-optimal solutions. Currently, the most popular method for minimizing the Fréchet function is the DTW Barycenter Averaging (DBA) algorithm proposed by Petitjean et al. [24]. Empirical results have shown that the DBA algorithm outperforms competing mean algorithms [24, 32].

This article proposes and investigates subgradient methods for minimizing the Fréchet function. The main contributions are as follows:

(i) A stochastic subgradient mean algorithm that outperforms DBA for non-small sample sizes.

(ii) Necessary and sufficient conditions of optimality.

(iii) Finite convergence of DBA to solutions satisfying the necessary conditions of optimality.

Subgradient methods are nonsmooth optimization [2] techniques that operate very similar to gradient descent methods, but replace the gradient with a subgradient. The concept of subgradient serves to generalize gradients under mild conditions that hold for the Fréchet function. We propose two subgradient methods, the majorize-minimize mean (MM) algorithm and the stochastic subgradient mean (SSG) algorithm. We show that the MM algorithm is equivalent to the DBA algorithm. Formulating DBA as a nonsmooth optimization method following a majorize-minimize [14] principle considerably simplifies its analysis and comparison to SSG.

Both algorithms DBA and SSG are iterative methods that repeatedly update a candidate solution. The main difference between both algorithms is that DBA is a batch and SSG a stochastic optimization method: While the DBA algorithm computes an exact subgradient on the basis of all sample time series, the SSG algorithm estimates a subgradient of the Fréchet function on the basis of a single randomly picked sample time series. Consequently, every time the algorithms have processed the entire sample, the SSG algorithm has performed N updates, whereas the DBA algorithm has performed a single update. The different update rules of both subgradient methods have the following implications:

1. Theoretical implication: The SSG algorithm is not a descent method. During optimization, the value of the Fréchet function can increase. In contrast, Petitjean et al. [27] proved that DBA is a descent method. Moreover, we show that DBA converges to solutions satisfying necessary conditions of optimality after a finite number of updates. Necessary conditions of optimality characterize the form of local minimizers of the Fréchet function. If a solution satisfies the sufficient conditions, we can conclude local minimality.

2. Practical implication: Empirical results suggest that the SSG algorithm is more stable and finds better solutions in substantially less time than the DBA algorithm provided that the sample size N is sufficiently large, that is $N > 50$ as a rule of thumb.

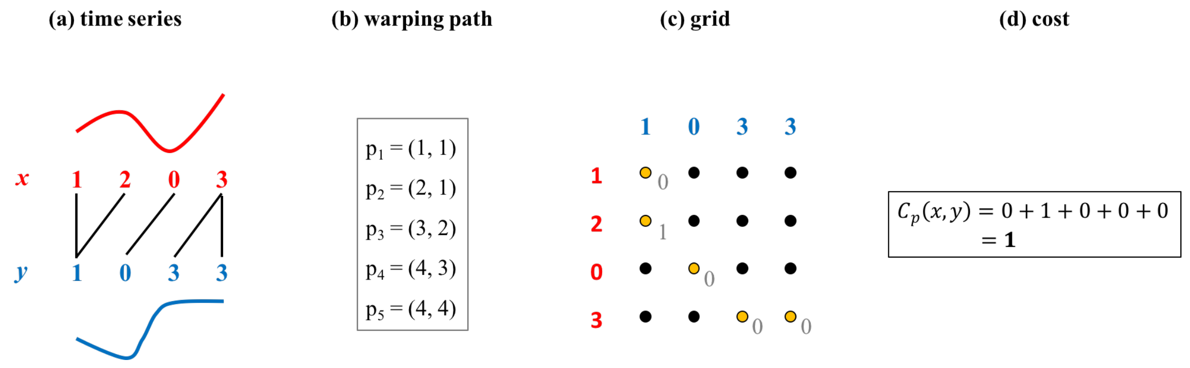


Figure 1: **(a)** Two time series x and y of length $n = 4$. The red and blue numbers are the elements of the respective time series. **(b)** Warping path $p = (p_1, \dots, p_5)$ of length $L = 5$. The points $p_l = (i_l, j_l)$ of warping path p align elements x_{i_l} of x to elements y_{j_l} of y as illustrated in (a) by black lines. **(c)** The 4×4 grid showing how warping path p moves from the upper left to the lower right corner as indicated by the orange balls. The numbers attached to the orange balls are the squared-error costs of the corresponding aligned elements. **(d)** The cost $C_p(x, y)$ of aligning x and y along warping path p.

In summary, the DBA algorithm has stronger theoretical properties than the SSG algorithm, whereas SSG exhibits superior performance than DBA for non-small sample sizes.

The rest of the article is structured as follows: Section 2 provides background material and discusses related work. In Section 3, we study analytical properties of the minimization problem. Section 4 proposes subgradient methods for minimizing the Fréchet function. In Section 5, we present and discuss empirical results. Finally, Section 6 concludes. Proofs are delegated to the appendix.

2. Background and Related Work

This section introduces the DTW distance, the Fréchet function, and finally reclassifies existing mean algorithms.

2.1. The Dynamic Time Warping Distance

We begin with introducing the DTW distance and refer to Figure 1 for illustrations of the concepts.

A *time series* x of *length* m is an ordered sequence $x = (x_1, \dots, x_m)$ consisting of *elements* $x_i \in \mathbb{R}^d$ for every *time point* $i \in [m]$, where we denote $[m] = \{1, \dots, m\}$. A time series is said to be *univariate* if $d = 1$ and *multivariate* if $d > 1$. For a fixed $d \in \mathbb{N}$, we denote by $\mathcal{T}_*$ the set of all time series of finite length and by $\mathcal{T}_m$ the set of all time series of length $m \in \mathbb{N}$. The DTW distance is a distance function on $\mathcal{T}_*$ based on the notion of warping path.

Definition 2.1. *Let $m, n \in \mathbb{N}$. A* warping path *of order $m \times n$ is a sequence $p = (p_1, \dots, p_L)$ of L points $p_l = (i_l, j_l) \in [m] \times [n]$ such that*

1. $p_1 = (1, 1)$ *and* $p_L = (m, n)$ (*boundary conditions*)
2. $p_{l+1} - p_l \in \{(1, 0), (0, 1), (1, 1)\}$ *for all* $l \in [L - 1]$ (*step condition*)

The set of all warping paths of order $m \times n$ is denoted by $\mathcal{P}_{m,n}$. A warping path of order $m \times n$ can be thought of as a path in a $[m] \times [n]$ grid, where rows are ordered top-down and columns are ordered left-right. The boundary condition demands that the path starts at the upper left corner and ends in the lower right corner of the grid. The step condition demands that a transition from point to the next point moves a unit in exactly one of the following directions: down, diagonal, and right.

A warping path $p = (p_1, \dots, p_L) \in \mathcal{P}_{m,n}$ defines an alignment (or warping) between time series $x = (x_1, \dots, x_m)$ and $y = (y_1, \dots, y_n)$. Every point $p_l = (i_l, j_l)$ of warping path p aligns element x_{i_l} to element y_{j_l}. The *cost* of aligning time series x and y along warping path p is defined by

$$C_p(x, y) = \sum_{l=1}^{L} \|x_{i_l} - y_{j_l}\|^2,$$

where the Euclidean norm $\|\cdot\|$ on $\mathbb{R}^d$ is used as *local cost function*. Then the DTW distance between two time series minimizes the cost of aligning both time series over all possible warping paths.

Definition 2.2. *Let x and y be two time series of length m and n, respectively. The* DTW distance *between x and y is defined by*

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

An optimal warping path *is any warping path $p \in \mathcal{P}_{m,n}$ satisfying* $\operatorname{dtw}(x, y) = \sqrt{C_p(x, y)}$.

A *DTW space* is any subset $\mathcal{T} \subseteq \mathcal{T}_*$ endowed with the DTW distance. In particular, $\mathcal{T}_*$ and $\mathcal{T}_m$ are DTW spaces. Even if the underlying local cost function is a metric, the induced DTW distance is generally only a pseudo-semi-metric satisfying

1. $\operatorname{dtw}(x, y) \geq 0$
2. $\operatorname{dtw}(x, x) = 0$

for all $x, y \in \mathcal{T}_*$. Computing the DTW distance and deriving an optimal warping path is usually solved by applying techniques from dynamic programming [29].

2.2. Fréchet Functions

Throughout this contribution, we consider a restricted form of the Fréchet function.

Definition 2.3. *Let $n \in \mathbb{N}$. Suppose that $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right)$ is a sample of N time series $x^{(k)} \in \mathcal{T}_*$. Then the function*

$$F : \mathcal{T}_n \to \mathbb{R}, \quad x \mapsto \frac{1}{N} \sum_{i=1}^{N} \operatorname{dtw}^2\left(x, x^{(k)}\right)$$

is the Fréchet function *of sample $\mathcal{X}$. The value $F(x)$ is the* Fréchet variation *of $\mathcal{X}$ at $x \in \mathcal{T}_n$.*

The domain $\mathcal{T}_n$ of the Fréchet function is restricted to the subset of all time series of fixed length n. In contrast, the sample time series $x^{(1)}, \dots, x^{(N)}$ can have arbitrary and different lengths.

Definition 2.4. *The* sample mean set *of $\mathcal{X}$ is the set*

$$\mathcal{F} = \{z \in \mathcal{T}_n \,:\, F(z) \leq F(x) \text{ for all } x \in \mathcal{T}_n\}.$$

Each element of $\mathcal{F}$ is called (sample) mean *of $\mathcal{X}$.*

A sample mean is a time series that minimizes the Fréchet function F. A sample mean of $\mathcal{X}$ exists [16], but is not unique, in general. We refer to the problem of minimizing the Fréchet function as the sample mean problem.

2.3. Related Work

The majority of time series averaging methods reported in the literature can be classified into two categories: (1) asymmetric-batch methods, and (2) symmetric-incremental methods. The asymmetric-symmetric dimension determines the type of average and the batch-incremental dimension determines the strategy with which a sample of time series is synthesized to one of both types of averages. We neither found work on symmetric-batch nor on asymmetric-incremental methods. There is a well-founded explanation for the absence of symmetric-batch algorithms, whereas asymmetric-incremental algorithms apparently have not been considered as a possible alternative to existing methods. A third category that has scarcely been applied to time series averaging are meta-heuristics. An exception are genetic algorithms proposed by Petitjean et al. [25]. In this section, we discuss existing methods within the symmetric-asymmetric and batch-incremental dimensions.

2.3.1. Asymmetric vs. Symmetric Averages

The asymmetric-symmetric dimension defines the form of an average. To describe the different forms of an average, we assume that p is an optimal warping path between time series x and y.

Asymmetric Averages. An asymmetric average of x and y is computed as follows: (i) select a reference time series, say x; (ii) compute an optimal warping path p between x and y; and (iii) average x and y along path p with respect to the time axis of reference x.

The different variants of the third step [1, 24, 28] have the following properties in common [18]: (i) they admit simple extension to averaging several time series; and (ii) the length of the resulting average coincides with the length of the reference x. Therefore, the form of an asymmetric average depends on the choice of the reference time series. Hence, an asymmetric average for a given optimal warping path between two time series is not well-defined, because there is no natural criterion for choosing a reference.

Symmetric Averages. According to Kruskal and Liberman [18], the symmetric average of time series x and y along an optimal warping path p is a time series z of the same length as warping path p and consists of elements $z_l = \left(x_{i_l} + y_{j_l}\right)/2$ for all points $p_l = (i_l, j_l)$ of p. Not only the elements x_i and y_j at the warped time points i and j but also the warped time points i and j themselves can be averaged.

Symmetric averages z of time series x and y are well-defined for a given optimal warping path p, but generally have more time points (a finer sampling rate) than x and y. Kruskal and Liberman [18] pointed to different methods for adjusting the sampling rate of z to the sampling rates of x and y, which results in averages whose length is adapted to the length of x and y.

2.3.2. Batch vs. Incremental Averaging

The batch-incremental dimension describes the strategy for combining more than two time series to an average.

Batch Averaging. Batch averaging first warps all sample time series into an appropriate form and then averages the warped time series. We distinguish between asymmetric-batch and symmetric-batch methods.

The *asymmetric-batch* method presented by Lummis [17] and Rabiner & Wilpon [28] first chooses an initial time series z as reference. Then the following steps are repeated until termination: (i) warp all time series onto the time axis of the reference z; and (ii) assign the average of the warped time series as new reference z. By construction, the length of the reference z is identical in every iteration. Consequently, the final solution (reference) of the asymmetric-batch method depends on the choice of the initial solution.

Since the early work in the 1970ies, different variants of the asymmetric-batch method have been proposed and applied. Oates et al. [22] and Abdulla et al. [1] applied an asymmetric-batch method confined to a single iteration. Hautamaki et al. [13] completed the approach by [1] by iterating the update step several times. With the DBA algorithm, Petitjean et al. [24] presented a sound solution in full detail. In the same spirit, Soheily-Khah et al. [33] generalized the asymmetric-batch method to weighted and kernelized versions of the DTW distance.

The *symmetric-batch* method suggested by Kruskal and Liberman [18] requires to find an optimal warping in an N-dimensional hypercube, where N is the sample size. Since finding an optimal common warping path for N sample time series is computationally intractable, symmetric-batch methods have not been further explored.

Incremental Averaging. Incremental methods synthesize several sample time series to an average time series by pairwise averaging. The general procedure repeatedly applies the following steps: (i) select a pair of time series x and y; (ii) compute average z of time series x and y; (iii) include z into the sample; (iv) optionally remove x and/or y from the sample. Incremental methods differ in the strategy of selecting the next pair of time series in step (i) and removing time series from the sample in step (iv). Pairwise averaging can take either of both forms, asymmetric and symmetric.

Kruskal and Liberman [18] presented a general description of symmetric-incremental methods in 1983 that includes progressive approaches as applied in multiple sequence alignment in bioinformatics [12]. The most cited concrete realizations of the Kruskal-Liberman method are weighted averages for self-organizing maps [34], two *nonlinear alignment and averaging filters* (NLAAF) [11], and the *prioritized shape averaging* (PSA) algorithm [19]. For further variants of incremental methods we refer to [20, 23, 35].

Finally, we note that there is apparently no work on asymmetric-incremental methods. The proposed SSG algorithm fills this gap. As we will see later, empirical results suggest to transform existing symmetric-incremental to asymmetric-incremental methods.

2.3.3. Empirical Comparisons

In experiments, the asymmetric-batch method DBA outperformed the symmetric-incremental methods NLAAF and PSA [24, 32]. This result shifted the focus from symmetric-incremental to asymmetric-batch methods. Since symmetric-batch methods are considered as computationally intractable, the open question is how asymmetric-incremental methods will behave and perform. This question is partly answered in this article.

3. Analytical Properties of the Sample Mean Problem

This section first analyzes the local properties of Fréchet functions. Then we present necessary and sufficient conditions of optimality. For the sake of clarity, the main text derives all results for the special case of univariate time series of fixed length n. The general case of multivariate sample time series of variable length is considered in Section B. Therefore, we use the following uncluttered notation in this and the following sections:

$\mathcal{T}$	:	*set of univariate time series* $x = (x_1, \dots, x_n)$ *of fixed length* $n \in \mathbb{N}$*, where* $x_i \in \mathbb{R}$
$\mathcal{T}^N$	:	N*-fold Cartesian product, where* $N \in \mathbb{N}$
$\mathcal{P}$	:	*set* $\mathcal{P}_{n,n}$ *of all warping paths of order* $n \times n$
$\mathcal{P}_*(x, y)$	:	*set of optimal warping paths between time series* x *and* y

3.1. Decomposition of the Fréchet Function

A discussion of analytical properties of the Fréchet function is difficult since standard analytical concepts such as locality, continuity, and differentiability are unknown in DTW spaces. To approach the sample mean problem analytically, we change the domain of the Fréchet function F from the DTW space $\mathcal{T}$ to the Euclidean space $\mathbb{R}^n$. This modification does not change the sample mean set, but local properties of F on different domains may differ. By saying the Fréchet function $F : \mathcal{T} \to \mathbb{R}$ has some analytical property, we tacitly assume its Euclidean characterization.

We decompose the Fréchet function into a mathematically more convenient form. Suppose that $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right) \in \mathcal{T}^N$ is a sample of time series. Substituting the definition of the DTW distance into the Fréchet function of $\mathcal{X}$ gives

$$F(x) = \frac{1}{N} \sum_{k=1}^{N} \text{dtw}^2\Big(x, x^{(k)}\Big) = \frac{1}{N} \sum_{k=1}^{N} \min_{p^{(k)} \in \mathcal{P}} \; C_{p^{(k)}}\Big(x, x^{(k)}\Big), \tag{1}$$

where $C_p(x, y)$ is the cost of aligning time series x and y along warping path p. Interchanging summation and the min-operator in Eq. (1) yields

$$F(x) = \min_{p^{(1)} \in \mathcal{P}} \cdots \min_{p^{(N)} \in \mathcal{P}} \frac{1}{N} \sum_{k=1}^{N} C_{p^{(k)}}\Big(x, x^{(k)}\Big). \tag{2}$$

To simplify the notation in Eq. (2), we introduce configurations and component functions. A *configuration* of warping paths is an ordered list $\mathcal{C} = \big(p^{(1)}, \ldots, p^{(N)}\big) \in \mathcal{P}^N$, where warping path $p^{(k)}$ is associated with time series $x^{(k)}$, $k \in [N]$. A *component function* of $F(x)$ is a function of the form

$$F_{\mathcal{C}} : \mathbb{R}^n \to \mathbb{R}, \quad x \mapsto \frac{1}{N} \sum_{k=1}^{N} C_{p^{(k)}}\Big(x, x^{(k)}\Big),$$

where $\mathcal{C} = \big(p^{(1)}, \ldots, p^{(N)}\big)$ is a configuration. Using the notions of configuration and component function, we can equivalently rewrite the Fréchet function as

$$F(x) = \min_{\mathcal{C} \in \mathcal{P}^N} F_{\mathcal{C}}(x). \tag{3}$$

We say, $F_{\mathcal{C}}$ is *active* at x if $F_{\mathcal{C}}(x) = F(x)$. By $\mathcal{A}_F(x)$ we denote the set of active component functions of F at time series x. A configuration $\mathcal{C}$ is *optimal* at x if $F_{\mathcal{C}}$ is an active component at x. In this case, every $p^{(k)} \in \mathcal{C}$ is an optimal warping path between x and $x^{(k)}$.

3.2. Local Lipschitz Continuity of the Fréchet Function

By definition of the cost functions C_p we find that every component function $F_{\mathcal{C}}$ is convex and differentiable. Thus, Eq. (3) implies that the Fréchet function F is the pointwise minimum of finitely many convex differentiable functions. Since convexity is not closed under min-operations, the Fréchet function is non-convex. Similarly, the Fréchet function F is not differentiable, because differentiability is also not closed under min-operations.

We show that the Fréchet function F is locally Lipschitz continuous.[1] Local Lipschitz continuity of F follows from two properties: (i) continuously differentiable functions are locally Lipschitz; and (ii) the local Lipschitz property is closed under the min-operation.

Any locally Lipschitz function is differentiable almost everywhere by Rademacher's Theorem [9]. In addition, locally Lipschitz functions admit a concept of generalized gradient at non-differentiable points, called subdifferential henceforth. The subdifferential $\partial F(x)$ of the Fréchet function F at point x is a non-empty, convex, and compact set. At differentiable points x, the subdifferential $\partial F(x)$ coincides with the gradient $\nabla F(x)$, that is $\partial F(x) = \{\nabla F(x)\}$. At non-differentiable points x, we have

$$\nabla F_{\mathcal{C}}(x) \in \partial F(x)$$

for all active component functions $F_{\mathcal{C}} \in \mathcal{A}_F(x)$. The elements $g \in \partial F(x)$ are called the subgradients of F at x. We refer to [2] for a definition of subdifferential for locally Lipschitz functions. Subdifferentials and subgradients have been originally defined for non-differentiable convex functions and later extended to locallly Lipschitz continuous functions by Clarke [6]. Throughout this article, we assume Clarke's definition of subdifferential and subgradient.

We conclude this section with introducing critical points. A point $x \in \mathcal{T}$ is called *critical* if $0 \in \partial F(x)$. Examples of critical points are the global minimizers of active component functions. Figure 2 depicts an example of a Fréchet function and some of its critical points.

[1]A function $f : \mathbb{R}^n \to \mathbb{R}$ is locally Lipschitz at point $x \in \mathbb{R}^n$ if there are scalars $L > 0$ and $\varepsilon > 0$ such that $|f(y) - f(z)| \leq L\,\|y - z\|$ for all $y, z \in \mathcal{B}(x, \varepsilon) = \{u \in \mathbb{R}^n \,:\, \|u - x\| \leq \varepsilon\}$.

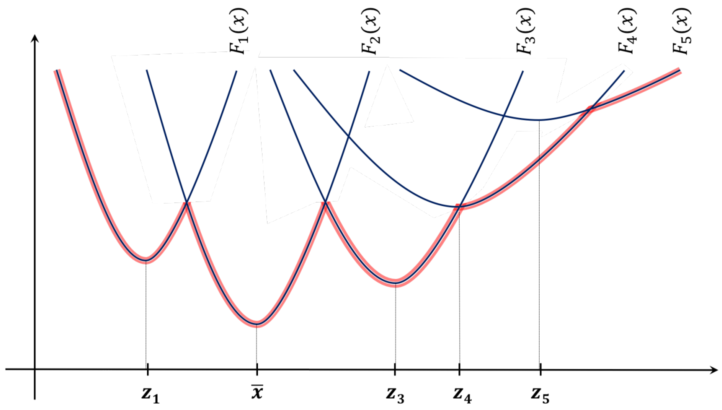


Figure 2: Schematic depiction of Fréchet function F of a sample $\mathcal{X} \in \mathcal{T}^N$ as pointwise minimum of the component functions $F_1, \dots, F_5$ indexed by integers rather than configurations from $\mathcal{P}^N$. The component functions are shown by blue lines and the Fréchet function is depicted by the red line. The red line also shows the surface of the points at which the respective component functions are active. Though the component functions are convex and differentiable, the Fréchet function is neither convex nor differentiable but differentiable almost everywhere. The minimizer of component function F_2 is the unique global minimizer of F and therefore the unique mean $\bar{x}$ of sample $\mathcal{X}$. The minimizers z_1 and z_3 of the component functions F_1 and F_3, resp., are local minimizers of F. The minimizer z_4 of component function F_4 is a non-differentiable critical point of F. Finally, the minimizer z_5 of component function F_5 is neither a local minimizer nor a non-differentiable critical point of F, because F_5 is not active at z_5.

3.3. Subgradients of the Fréchet Function

This section aims at describing an arbitrary subgradient of F for each point $x \in \mathcal{T}$. As discussed in Section 3.2 the gradient of an active component function $F_\mathcal{C}$ at x is a subgradient of F at x. Since for each $x \in \mathcal{T}$ there is an active component function $F_\mathcal{C} \in \mathcal{A}_F(x)$, it is sufficient to specify gradients of component functions. For this, we introduce the notions of warping and valence matrix.

Definition 3.1. *Let $p \in \mathcal{P}$ be a warping path.*

1. *The* warping matrix *of p is a matrix $W \in \{0,1\}^{n \times n}$ with elements*

$$W_{i,j} = \begin{cases} 1 & : \quad (i,j) \in p \\ 0 & : \quad \textit{otherwise} \end{cases}.$$

2. *The* valence matrix *of p is the diagonal matrix $V \in \mathbb{N}^{n \times n}$ with integer elements*

$$V_{i,i} = \sum_{j=1}^{n} W_{i,j}.$$

Figure 3 provides an example of a warping and valence matrix. The warping matrix is a matrix representation of its corresponding warping path. The valence matrix is a diagonal matrix, whose elements count how often an element of the first time series is aligned to an element of the second one.

The next result describes the gradients of a component function.

Proposition 3.2. *Let $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right) \in \mathcal{T}^N$ be a sample of time series and let $\mathcal{C} \in \mathcal{P}^N$ be a configuration of warping paths. The gradient of component function $F_\mathcal{C}$ at $x \in \mathcal{T}$ is of the form*

$$\nabla F_\mathcal{C}(x) = \frac{2}{N} \sum_{k=1}^{N} \left(V^{(k)} x - W^{(k)} x^{(k)}\right), \tag{4}$$

where $V^{(k)}$ and $W^{(k)}$ are the valence and warping matrix of warping path $p^{(k)} \in \mathcal{C}$ associated with time series $x^{(k)} \in \mathcal{X}$.

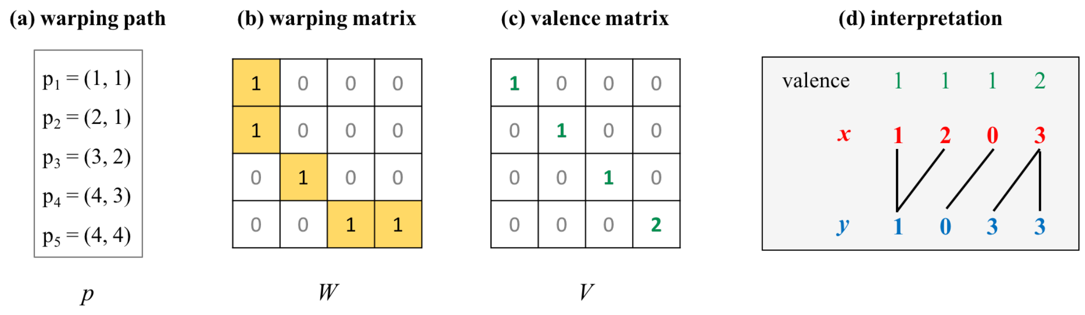


Figure 3: Illustration of warping and valence matrix. Box (a) shows the warping path p of Figure 1. Box (b) shows the warping matrix W of warping path p. The points $p_l = (i_l, j_l)$ of warping path p determine the ones in the warping matrix W. Box (c) shows the valence matrix V of warping path p. The matrix V is a diagonal matrix, whose elements $V_{i,i}$ are the row sums of W. Box (d) interprets the valence matrix V. The valence $V_{i,i}$ of element x_i of time series x is the number of elements in time series y that are aligned to x_i by warping path p. Thus, the valence $V_{i,i}$ is the number of black lines emanating from element x_i.

3.4. Necessary and Sufficient Conditions of Optimality

The sample mean problem is posed as an unconstrained optimization problem. The classical mathematical approach to an optimization problem studies the four basic questions: (i) the formulation of necessary conditions, (ii) the formulation of sufficient conditions, (iii) the question of the existence of solutions, and (iv) the question of the uniqueness of a solution. Existence of a sample mean has been proved [16] and non-uniqueness follows by constructing examples. In this section, we answer the remaining two questions (i) and (ii).

The next theorem presents the necessary conditions of optimality in terms of warping and valence matrices. For an illustration of the theorem, we refer to Figure 4.

Theorem 3.3. *Let F be the Fréchet function of sample $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right) \in \mathcal{T}^N$. If $z \in \mathcal{T}$ is a local minimizer of F, then there is a configuration $\mathcal{C} \in \mathcal{P}^N$ such that the following conditions are satisfied:*

(C1) *$F(z) = F_{\mathcal{C}}(z)$.*

(C2) *We have*

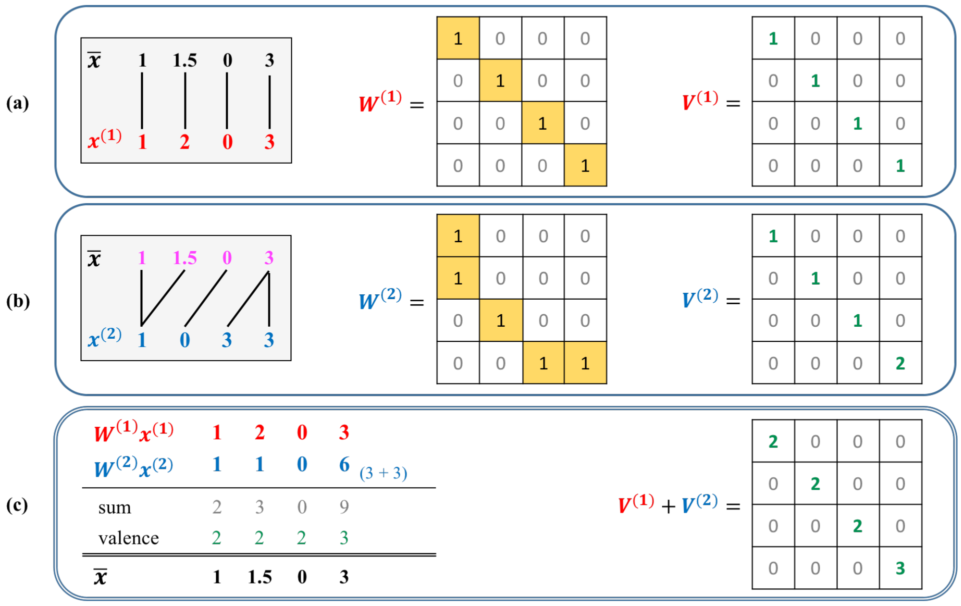
$$z = \left(\sum_{k=1}^{N} V^{(k)}\right)^{-1} \left(\sum_{k=1}^{N} W^{(k)}\, x^{(k)}\right), \tag{5}$$

where $V^{(k)}$ and $W^{(k)}$ are the valence and warping matrix of $p^{(k)} \in \mathcal{C}$ for all $k \in [N]$.

Since (C1) and (C2) are necessary but not sufficient conditions, there are time series that satisfy both conditions but are not local minimizers of the Fréchet function. Condition (C2) implies that z is the unique minimizer of the component function $F_{\mathcal{C}}$ (cf. A.3). As illustrated in Figure 2, three cases can occur: (i) z is a local minimizer of F, (ii) z is a non-differentiable critical point, and (iii) z is neither a local minimizer nor a non-differentiable critical point. Condition (C1) eliminates solutions of the third case, which are minimizers of inactive component functions. Hence, solutions satisfying the necessary conditions of optimality include local minimizers and non-differentiable critical point. The second result of this section presents a sufficient condition of a local minimum.

Proposition 3.4. *Let F be the Fréchet function of sample $\mathcal{X}$ and let $z \in \mathcal{T}$ be a time series. Suppose that there is a configuration $\mathcal{C} \in \mathcal{P}^N$ such that z satisfies the necessary conditions (C1) and (C2). If $\mathcal{C}$ is unique, then z is a local minimizer.*

Uniqueness of configuration $\mathcal{C}$ in Prop. 3.4 is equivalent to the statement that the active set at z is of the form $\mathcal{A}_F(z) = \{F_{\mathcal{C}}\}$. Thus, the sufficient condition of Prop. 3.4 is difficult to verify. One naive



$$W^{(1)} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}, \quad V^{(1)} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&1\end{pmatrix}$$

$$W^{(2)} = \begin{pmatrix}1&0&0&0\\1&0&0&0\\0&1&0&0\\0&0&1&1\end{pmatrix}, \quad V^{(2)} = \begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&2\end{pmatrix}$$

$$V^{(1)} + V^{(2)} = \begin{pmatrix}2&0&0&0\\0&2&0&0\\0&0&2&0\\0&0&0&3\end{pmatrix}$$

Figure 4: Mean $\bar{x}$ of time series $x^{(1)}$ and $x^{(2)}$. The two boxes (a) and (b) show the warping matrix $W^{(k)}$ and the valence matrix $V^{(k)}$ of the optimal warping path between mean $\bar{x}$ and time series $x^{(k)}$ for $k \in \{1, 2\}$. Box (c) shows computation of the mean $\bar{x}$ using the warping and valence matrices. The matrix on the right hand side shows the sum $V = V^{(1)} + V^{(2)}$ of both valence matrices. The first two lines on the left hand side show the results of the matrix multiplication $y^{(k)} = W^{(k)} \cdot x^{(k)}$. The third line shows the sum $y = y^{(1)} + y^{(2)}$. The fourth line shows the valences corresponding to the diagonal elements of V. The valences count how many elements of the sample time series are aligned to a given element of the mean. The last line shows the mean $\bar{x}$ obtained by element-wise division of the sum by the valences. In matrix notation, the mean is given by $\bar{x} = V^{-1}y$.

way to test the condition is to enumerate all optimal configurations at z. If there is exactly one optimal configuration, then z is a local minimizer.

We conclude this section with a discussion on the form of a mean. Any sample mean is of the form of Eq. (5), because global minimizers are also local minimizers. As expected, a sample mean in DTW spaces generalizes the arithmetic mean in Euclidean spaces. The term in Eq. (5) is a normalized sum of (transformed) sample time series. Intuitively speaking, the difference to the arithmetic mean is that the sample time series are aligned by warping transformations and the normalization factor becomes a diagonal matrix (inverted sum of the valence matrices) which normalizes each element of the time series separately. Indeed, if every warping matrix is the identity matrix, then Eq. (5) is the sample mean of $\mathcal{X}$ in the Euclidean sense.

4. Mean Algorithms

In the previous section, we have shown that Fréchet functions are locally Lipschitz continuous. For such functions, the field of nonsmooth analysis has developed several optimization methods [2]. One of the most simplest nonsmooth optimization techniques are subgradient methods originally developed for minimization of convex functions [31]. This section adopts three (generalized) subgradient methods for minimizing the Fréchet function: (1) the subgradient mean algorithm, (2) the majorize-minimize mean algorithm, and (3) the stochastic version of the subgradient mean algorithm. We provide MATLAB and Java implementations of the last two algorithms at [30].

Algorithm 1 Subgradient Mean Algorithm

```
 1: procedure SG($x^{(1)}, \dots, x^{(N)}$)
 2:     initialize solution $z \in \mathcal{T}$
 3:     initialize best solution $z_* = z$
 4:     repeat
 5:         for all $k \in [N]$ do (optionally in parallel threads)
 6:             compute optimal warping path $p^{(k)} \in \mathcal{P}_*\big(z, x^{(k)}\big)$ between $z$ and $x^{(k)}$
 7:             derive valence matrix $V^{(k)}$ and warping matrix $W^{(k)}$ of $p^{(k)}$
 8:         end for
 9:         update solution $z$ according to the rule
                $z \leftarrow z - \eta \, \frac{2}{N} \sum_{k=1}^{N} \Big( V^{(k)} z - W^{(k)} x^{(k)} \Big)$
10:         update best solution $z_*$ such that $F(z_*) = \min\{F(z_*), F(z)\}$
11:         adjust step size $\eta$
12:     until termination
13:     return $z_*$ as approximation of a mean
14: end procedure
```

4.1. The Subgradient Mean Algorithm

Subgradient methods for minimizing a locally Lipschitz continuous function $f(x)$ look similar to gradient-descent algorithms. The update rule of the standard subgradient method is of the form

$$z^{(t+1)} = z^{(t)} - \eta^{(t)} \cdot g^{(t)}, \tag{6}$$

where $z^{(t)}$ is the solution at iteration t, $\eta^{(t)}$ is the t-th step size, and $g^{(t)} \in \partial f(z^{(t)})$ is an arbitrary subgradient of f at $z^{(t)}$. Since the standard subgradient method is not a descent method, it is common to keep track of the best solution $z_*^{(t)}$ found so far, that is

$$f\left(z_*^{(t)}\right) = \min\left\{ f\left(z_*^{(t-1)}\right), f\left(z^{(t)}\right) \right\}.$$

with $z_*^{(0)} = z^{(0)}$. We apply this idea to the sample mean problem by using the gradient of an arbitrary active component function (cf. Prop. 3.2) as subgradient of the Fréchet function F at the current solution $z^{(t)}$.

Algorithm 1 outlines the basic procedure of the subgradient mean (SG) algorithm for minimizing the Fréchet function $F(x)$ of a sample $\mathcal{X} = \big(x^{(1)}, \dots, x^{(N)}\big) \in \mathcal{T}^N$ of time series. The SG algorithm consists of the following steps:

1. *Initialize*: The performance of the SG algorithm depends on the choice of initial solution. Efficient and simple strategies to initialize the solution in `line 2` are randomly selecting a sample time series $x^{(k)}$ or generating a random time series. Several authors independently suggested to select a medoid time series from $\mathcal{X}$ [13, 22, 27], which requires $\mathcal{O}(N^2)$ computations of the DTW distance. For large sample size N, selecting a medoid of $\mathcal{X}$ is computationally infeasible. An efficient alternative is to select a medoid of a randomly chosen sub-sample of $\mathcal{X}$.

2. *Align sample time series*: In `line 5-8`, the SG algorithm cycles through all time series of the sample $\mathcal{X}$ and determines a configuration of optimal warping paths $p^{(k)}$ between the current solution z and the sample time series $x^{(k)}$. Next, the valence and warping matrix of the optimal warping paths $p^{(k)}$ are derived. This step implicitly determines an active component function $F_{\mathcal{C}}$ at the current solution z. It is the computationally most demanding part of the algorithm, but can be easily executed in N parallel threads, where each thread executes `line 6-7` for a different $k \in [N]$.

3. *Update solution*: `Line 9` implements the update rule of the standard subgradient method corresponding to Eq. (6), where the subgradient corresponds to the gradient of the active component function $F_{\mathcal{C}}$.

4. *Update best solution*: `Line 10` keeps track of the best solution z_* found so far. Updating the best solution evaluates the Fréchet function $F(z)$ at the new solution z, which requires N computations of the

Algorithm 2 Majorize-Minimize Mean Algorithm

```
 1: procedure MM($x^{(1)}, \dots, x^{(N)}$)
 2:     initialize solution $z \in \mathcal{T}$
 3:     repeat
 4:         //*** Majorization *******************************************************//
 5:         for all $k \in [N]$ do (optionally in parallel threads)
 6:             compute optimal warping path $p^{(k)} \in \mathcal{P}_*\big(z, x^{(k)}\big)$ between $z$ and $x^{(k)}$
 7:             derive valence matrix $V^{(k)}$ and warping matrix $W^{(k)}$ of $p^{(k)}$
 8:         end for
 9:         //*** Minimization *******************************************************//
10:         update solution $z$ according to the rule
                $$z \leftarrow \left(\sum_{k=1}^{N} V^{(k)}\right)^{-1} \left(\sum_{k=1}^{N} W^{(k)} x^{(k)}\right)$$
11:     until $F(z)$ is equal to the Fréchet variation of the previous iteration
12:     return $z$
13: end procedure
```

DTW distance. To keep the computational effort low, the resulting optimal warping paths are used in the alignment step 2. of the next iteration. Consequently, this step consumes no substantial additional computational resources, except at the last iteration.

5. *Adjust step size*: In `line 11`, the step size is adjusted according to some schedule. There are several schedules for adapting the step size including the special case of a constant step size. The choice of a schedule for adjusting the step size affects the convergence behavior of the algorithm.

6. *Terminate*: A good termination criterion is important, because a premature termination by a weak criterion may result in a useless solution, whereas a too severe criterion may result in an iteration process that is computationally infeasible. We suggest the following two termination criteria: (T1) Terminate when a maximum number of iterations has been exceeded; and (T2) terminate when no improvement is observed after some iterations. Improvement in (T2) is measured by the evaluations of the Fréchet function in `line 10`.

We conclude this section by placing the SG algorithm into the context of existing mean algorithms. As the DBA algorithm, the SG algorithm falls into the class of asymmetric-batch methods (cf. Section 2.3). The difference between existing asymmetric-batch methods and the SG algorithm is the way with which the sample time series are weighted when synthesized to an average. Existing asymmetric-batch approaches use a fixed weighting scheme, whereas the SG algorithm admits a more general and flexible weighting scheme based on adaptive step sizes. In Section 4.2 we show that this flexibility includes the DBA algorithm. This means, DBA is a special case of the SG algorithm.

4.2. The Majorize-Minimize Mean Algorithm

The second mean algorithm exploits the necessary conditions of optimality stated in Theorem 3.3 and belongs to the class of majorize-minimize algorithms [14]. This class includes the EM algorithm as special case and provides access to general convergence results [36]. Algorithm 2 outlines the basic procedure of the majorize-minimize mean (MM) algorithm. After initialization, the MM algorithm repeatedly alternates between majorization and minimization until convergence (cf. Figure 5):

Majorization. A real-valued function $G : \mathcal{T} \to \mathbb{R}$ majorizes the Fréchet function F at time series $z \in \mathcal{T}$ if $F(z) = G(z)$ and $F(x) \leq G(x)$ for all time series $x \in \mathcal{T}$. Typically, one chooses a majorizing function that is much easier to minimize than the original function. For example, any active component function $F_{\mathcal{C}} \in \mathcal{A}_F(z)$ majorizes the Fréchet function F at z. `Line 5-8` implicitly determine such a component function $F_{\mathcal{C}}$ by constructing a configuration $\mathcal{C}$ of optimal warping paths $p^{(k)} \in \mathcal{P}_*\big(z, x^{(k)}\big)$ between the current solution z and sample time series $x^{(k)}$.

Minimization. The minimization step in `line 10` computes the minimum of the majorizing function $F_{\mathcal{C}}$ as next solution of the iteration process. Since an active component function is convex and differentiable,

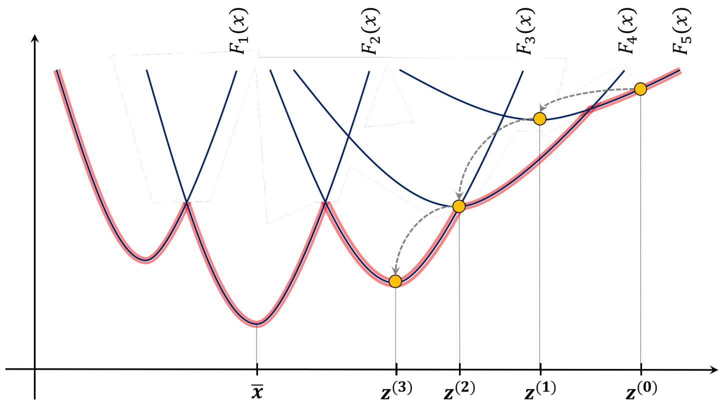


Figure 5: Illustration of the MM algorithm. The algorithm starts with the initial solution $z^{(0)}$. At each iteration t, majorization determines a configuration $\mathcal{C}^{(t)} \in \mathcal{P}^N$ of optimal warping paths between the current solution $z^{(t)}$ and the sample time series $x^{(k)}$. The configuration $\mathcal{C}^{(t)}$ determines the majorizing component function $F_{\mathcal{C}^{(t)}}$, which is active at the current solution $z^{(t)}$. Then the minimization-step determines the unique minimum x_* of the component function $F_{\mathcal{C}^{(t)}}$. The iteration continues with $z^{(t+1)} = x_*$ as updated solution.

its unique minimizer is easy to compute by setting the gradient $\nabla F_{\mathcal{C}}(z)$ given in Eq. (4) to zero and solving for z. The resulting minimizer takes the form of the necessary condition (C2) stated in Theorem 3.3.

Majorize-Minimize algorithms are descent methods by construction. Hence, if $(z^{(t)})$ is a sequence of points updated at iteration t of the MM algorithm, then the sequence $(F(z^{(t)}))$ of Fréchet variations at the updated points $z^{(t)}$ is monotonously decreasing. The next theorem shows finite convergence of the MM algorithm to a solution satisfying necessary conditions of optimality.

Theorem 4.1. *The MM algorithm terminates after a finite number of iterations at a solution satisfying the necessary conditions of optimality (C1) and (C2).*

One way to prove Theorem 4.1 is to invoke Zangwill's convergence theorem. Checking the assumptions of Zangwill's convergence theorem is often as difficult as finding a direct proof [4]. Here, we present a direct proof of Theorem 4.1, which is a stronger convergence statement than we would obtain by invoking Zangwill's convergence theorem. Theorem 4.1 yields the termination criterion used in Algorithm 2:

Corollary 4.2. *Let $(z^{(1)}, z^{(2)}, \dots)$ be a sequence of updated points generated by the MM algorithm. If $F(z^{(t)}) = F(z^{(t-1)})$ for some iteration t, then $z^{(t)}$ satisfies the necessary conditions of optimality (C1) and (C2) stated in Theorem 3.3. Moreover, if the majorization step behaves deterministic in terms of the choice of optimal warping paths, then $z^{(t)}$ cannot be improved by the MM algorithm.*

We conclude this section with placing the MM algorithm into broader context. First, the MM algorithm is equivalent to the DBA algorithm proposed by Petitjean et al. [24]. The valence and warping matrix in `line 7` contain all the information gathered in the association tables of the DBA algorithm. The update rules of both algorithms are identical but expressed in different ways. The MM algorithm is preferred in programming languages that benefit from matrix and vector operations instead of loops. In addition, the MM algorithm is easily parallelizable in the same manner as the SG algorithm. We want to note that the descent property of DBA has been proved in [27]. Now, in context of majorize-minimize algorithms the descent property becomes self-evident.

Second, the mean algorithms proposed by Soheily-Khah et al. [33] use the same majorize-minimize optimization scheme as the MM algorithm for weighted and kernelized DTW distances. Their mean algorithm for the weighted DTW distance can be directly obtained by adapting the update rule in Algorithm 2 corresponding to the subgradient of the Fréchet function with weighted DTW distance. For the kernelized DTW distance we have a maximization problem. In this case they use minorize-maximize optimization. Once an active component function is selected in the minorization step, they perform ordinary gradient acent on the selected component function in the maximization step. Therefore, the mean algorithms in [33] generalize DBA to weighted and kernelized DTW distances.

Third, the MM algorithm is closely related to the corresponding MM algorithms for computing a mean of graphs [15] and a mean partition in consensus clustering [7]. All three algorithms have in common that the underlying Fréchet function is a pointwise minimum of convex differentiable component functions.

Fourth, the MM algorithm can be regarded as a special variant of a subgradient algorithm. By using the per-coordinate step size

$$\eta = \left(\frac{2}{N} \sum_{k=1}^{N} V^{(k)} \right)^{-1}, \tag{7}$$

the SG Algorithm 1 reduces to the MM algorithm. Observe that the step size in the standard SG algorithm is a positive-valued scalar. In contrast, the step size η given in Eq. (7) is a diagonal matrix that provides individual step sizes for each element (coordinate) of a time series.

4.3. The Stochastic Subgradient Mean Algorithm

The SG algorithm computes a subgradient on the basis of the complete sample of time series. In contrast, the stochastic subgradient mean (SSG) algorithm updates the current solution on the basis of a single randomly selected time series $x^{(k)} \in \mathcal{X}$. The k-th term of the Fréchet function is of the form

$$F^{(k)}(x) = \text{dtw}^2\left(x, x^{(k)}\right) = \min_{p \in \mathcal{P}} C_p\left(x, x^{(k)}\right).$$

As an immediate consequence of Prop. 3.2, the gradient of component function $F_p^{(k)}(x) = C_p\left(x, x^{(k)}\right)$ is given by

$$\nabla F_p^{(k)}(x) = 2\Big(V^{(k)}x - W^{(k)}x^{(k)}\Big).$$

Algorithm 3 outlines the basic procedure of the stochastic subgradient mean algorithm for minimizing the Fréchet function of a sample $\mathcal{X} = \left(x^{(1)}, \dots, x^{(N)}\right) \in \mathcal{T}^N$ of time series. The individual steps of the SSG and SG algorithm are similar to a large extent, yet a few points deserve further explanation:

1. *Online setting*: Algorithm 3 presents the SSG algorithm as an incremental version of the SG algorithm that cycles several times through a given sample of size N. The SSG algorithm can also be applied in an online setting, where a stationary source with limited memory generates a sequence of time series. In this case, the SSG algorithm receives the next sample time series, updates the current solution, and then discards the current sample time series. There is no updating of the best solution. We can use (T1) as termination criterion (see discussion of the SG algorithm). When viewed as a stochastic optimization problem, we refer to [8, 21] for convergence results.

2. *Iteration*: Here, the number of iterations refers to the number of update steps. The SG algorithm considers the entire sample in one iteration, whereas the SSG algorithms considers a single sample time series in one iteration.

3. *Update best solution*: Recall that keeping track of the best solution z_* in `line 9` evaluates the Fréchet function $F(z)$, which requires N computations of the DTW distance. While the SG algorithm can use the resulting optimal warping paths for deriving the valence and warping matrices in the next iteration, the SSG algorithm can reuse only a single optimal warping path. Thus, one cycle through the entire sample by the SGG algorithm requires nearly twice as many DTW distance computations as the SG algorithm. To reduce the computational cost, we can either omit this step or apply it in regular intervals.

4. *Terminate*: The discussion about additional computational cost in item (3) carries over to termination criterion (T2).

We conclude this section by classifying the SSG algorithm into the asymmetric-symmetric and batch-incremental dimension as described in Section 2.3. The SSG algorithm implements incremental averaging by following a similar scheme as the symmetric-incremental method NLAAF2 [11]. In contrast to NLAAF2, the type of average in SGG is asymmetric. Thus, SSG is the first proposed mean algorithm belonging to the class of asymmetric-incremental methods.

Algorithm 3 Stochastic Subgradient Method

```
1: procedure SSG($x^{(1)}, \dots, x^{(N)}$)
2:     initialize solution $z \in \mathcal{T}$
3:     initialize best solution $z_* = z$
4:     repeat
5:         randomly select $k \in [N]$
6:         compute optimal warping path $p^{(k)} \in \mathcal{P}_*\big(z, x^{(k)}\big)$ between $z$ and $x^{(k)}$
7:         derive valence matrix $V^{(k)}$ and warping matrix $W^{(k)}$ of $p^{(k)}$
8:         update solution $z$ according to the rule
               $z \leftarrow z - \eta \left(V^{(k)} z - W^{(k)} x^{(k)}\right)$
9:         update best solution $z_*$ such that $F(z_*) = \min\{F(z_*), F(z)\}$
10:        adjust step size $\eta$
11:    until termination
12:    return $z$
13: end procedure
```

5. Experiments

The goal of this section is to assess the performance of the SSG algorithm in comparison with the MM (DBA) algorithm.

5.1. General Performance Comparison

The first series of experiments compares the performance of different variants of SSG and MM on selected UCR benchmark datasets.

5.1.1. Data

In this experiment, the 24 datasets of the UCR Time Series Classification Archive [5] shown in Table 1 were selected. Time series of the same dataset have identical length. The training and test set of the original datasets were merged to a single set.

5.1.2. Algorithms

The following five variants of the SSG and MM algorithm were considered:

Notation	Algorithm	Epochs
SSG-1	stochastic subgradient mean algorithm	1
SSG-e	stochastic subgradient mean algorithm	$e \in [50]$
SSG-50	stochastic subgradient mean algorithm	50
MM-1	majorize-minimize mean algorithm	1
MM-50	majorize-minimize mean algorithm	50

One epoch is a cycle through the whole dataset. SSG-1 and MM-1 terminate after the first epoch and SSG-50 after 50 epochs. MM-50 terminates as described in Algorithm 2, but at the latest after 50 epochs. Finally, SSG-e terminates after the same number of epochs as MM-50.

The step size of the SSG algorithms were adjusted according to the following schedule:

$$\eta^{(t)} = \begin{cases} \eta^{(t-1)} - (\eta_0 - \eta_1)/N & : \quad 1 \leq t \leq N \\ \\ \eta_1 & : \quad t > N \end{cases},$$

where $\eta_0 = \eta^{(0)} = 0.05$ is the initial step size, $\eta_1 = 0.005$ is the final step size, t is the number of iterations, and N is the sample size. This schedule linearly decreases the step size from η_0 to η_1 during the first epoch and then remains constant at η_1 until termination.

Table 1: UCR datasets and their characteristics. Shown are the name of the dataset, the length n of the time series, the number C of classes, and the sample size N.

Dataset	n	C	N	Dataset	n	C	N
50words	270	50	905	Gun Point	150	2	200
Adiac	176	37	781	Lighting2	637	2	121
Beef	470	5	60	Lighting7	319	7	143
CBF	128	3	930	OliveOil	570	4	60
ChlorineConcentration	166	3	4307	OSULeaf	427	6	442
Coffee	286	2	56	PhalangesOutlinesCorrect	80	2	2658
ECG200	96	2	200	SwedishLeaf	128	15	1125
ECG5000	140	5	5000	synthetic control	60	6	600
ElectricDevices	96	7	16637	Trace	275	4	200
FaceAll	131	14	2250	Two Patterns	128	4	5000
FaceFour	350	4	112	wafer	152	2	7164
FISH	463	7	350	yoga	426	2	3300

The solution quality of an algorithm is measured by means of Fréchet variation of the found solution. We denote the variation of algorithm A at epoch e by

$$V_A(e) = F\left(z_*^{(e)}\right),$$

where F is the Fréchet function and $z_*^{(e)}$ is the best solution found so far by algorithm A.

5.1.3. Experimental Protocol

Given a dataset, an experiment was conducted according to the following procedure:

```
1. given: dataset $\mathcal{X}$
2. for 30 trials do
3.     randomly select initial solution $z \in \mathcal{X}$
4.     run SSG for 50 epochs with initial solution $z$
5.     record variation $V_{SSG}(e)$ for $e \in \{1, \dots, 50\}$
6.     run MM for 50 epochs with initial solution $z$
7.     record variation $V_{MM}(e)$ for $e \in \{1, \dots, 50\}$
8. end for
```

Thus, in total, 720 trials (24 datasets $\times$ 30 trials) were conducted.

5.1.4. Results

Table 2 summarizes the results obtained by the five mean algorithms. The results show the average variations over the 30 trials and their standard deviations. The best average variations vary from 0.03 for OliveOil to 78.05 for Lighting2 both obtained by SSG-50. These results suggest that the selected datasets cover a broad range of variability in the data.

We compare the solution quality of the five mean algorithms. The left plot of Figure 6 shows the matrix $\left(p_{ij}^w\right)$ of percentage of wins. The percentage p_{ij}^w of wins is the fraction of trials in which the algorithm in row i found a solution with lower variation than the algorithm in column j. Note that

$$p_{ij}^{eq} = 100 - p_{ij}^w - p_{ji}^w$$

is the percentage of trials for which algorithms i and j found solutions with identical variation. The right plot of Figure 6 shows the average percentage p_{ij}^v of change in variation over all 720 trials. The percentage p_{ij}^v of change in variation of a single trial is defined by

$$p_{ij}^v = 100 \cdot \frac{V_j - V_i}{V_j},$$

where V_i and V_j are the variations of the solutions found by algorithms in row i and column j, respectively. Positive (negative) values p_{ij}^v mean that the algorithm in row i won (lost) against the reference algorithm in column j. We made the following observations:

Table 2: Average variation (lower is better) and standard deviation over 30 trials.

Dataset	SSG-1	SSG-e	SSG-50	MM-1	MM-50
50words	18.29 $^{(\pm 0.57)}$	17.71 $^{(\pm 0.53)}$	17.56 $^{(\pm 0.48)}$	34.67 $^{(\pm 5.28)}$	18.68 $^{(\pm 1.93)}$
Adiac	0.55 $^{(\pm 0.01)}$	0.53 $^{(\pm 0.00)}$	0.52 $^{(\pm 0.00)}$	0.8 $^{(\pm 0.05)}$	0.54 $^{(\pm 0.01)}$
Beef	27.48 $^{(\pm 4.51)}$	19.95 $^{(\pm 2.41)}$	15.9 $^{(\pm 1.16)}$	38.96 $^{(\pm 6.68)}$	16.86 $^{(\pm 2.24)}$
CBF	18.35 $^{(\pm 0.34)}$	18.06 $^{(\pm 0.30)}$	18.01 $^{(\pm 0.31)}$	23.68 $^{(\pm 1.67)}$	18.4 $^{(\pm 0.35)}$
ChlorineConcentration	14.79 $^{(\pm 0.35)}$	14.71 $^{(\pm 0.36)}$	14.71 $^{(\pm 0.35)}$	17.64 $^{(\pm 1.13)}$	15.13 $^{(\pm 0.52)}$
Coffee	0.77 $^{(\pm 0.03)}$	0.7 $^{(\pm 0.02)}$	0.67 $^{(\pm 0.01)}$	0.91 $^{(\pm 0.08)}$	0.68 $^{(\pm 0.01)}$
ECG200	7.15 $^{(\pm 0.49)}$	7.15 $^{(\pm 0.49)}$	6.76 $^{(\pm 0.40)}$	8.81 $^{(\pm 0.78)}$	6.95 $^{(\pm 0.43)}$
ECG5000	17.91 $^{(\pm 0.99)}$	17.79 $^{(\pm 0.97)}$	17.76 $^{(\pm 0.97)}$	26.92 $^{(\pm 4.22)}$	19.21 $^{(\pm 1.58)}$
ElectricDevices	42.74 $^{(\pm 0.27)}$	42.54 $^{(\pm 0.22)}$	42.48 $^{(\pm 0.22)}$	56.84 $^{(\pm 4.99)}$	43.54 $^{(\pm 0.49)}$
FaceAll	27.6 $^{(\pm 0.39)}$	27.44 $^{(\pm 0.36)}$	27.4 $^{(\pm 0.34)}$	33.11 $^{(\pm 1.65)}$	28.25 $^{(\pm 0.76)}$
FaceFour	37.69 $^{(\pm 1.68)}$	35.83 $^{(\pm 1.37)}$	35.18 $^{(\pm 1.37)}$	48.54 $^{(\pm 4.87)}$	36.43 $^{(\pm 1.53)}$
FISH	1.33 $^{(\pm 0.03)}$	1.29 $^{(\pm 0.03)}$	1.29 $^{(\pm 0.02)}$	1.67 $^{(\pm 0.16)}$	1.33 $^{(\pm 0.08)}$
Gun_Point	2.72 $^{(\pm 0.34)}$	2.63 $^{(\pm 0.36)}$	2.41 $^{(\pm 0.29)}$	5.99 $^{(\pm 1.11)}$	2.4 $^{(\pm 0.20)}$
Lighting2	88.17 $^{(\pm 3.71)}$	82.34 $^{(\pm 2.28)}$	78.05 $^{(\pm 1.45)}$	116.92 $^{(\pm 9.30)}$	79.95 $^{(\pm 2.41)}$
Lighting7	52.73 $^{(\pm 1.97)}$	48.88 $^{(\pm 1.00)}$	48.59 $^{(\pm 0.94)}$	70.32 $^{(\pm 4.50)}$	49.51 $^{(\pm 1.20)}$
OliveOil	0.03 $^{(\pm 0.00)}$	0.03 $^{(\pm 0.00)}$	0.03 $^{(\pm 0.00)}$	0.03 $^{(\pm 0.00)}$	0.03 $^{(\pm 0.00)}$
OSULeaf	29.15 $^{(\pm 0.59)}$	29.15 $^{(\pm 0.59)}$	27.68 $^{(\pm 0.39)}$	51.72 $^{(\pm 8.67)}$	28.41 $^{(\pm 0.71)}$
PhalangesOutlinesCorrect	1.17 $^{(\pm 0.01)}$	1.16 $^{(\pm 0.01)}$	1.16 $^{(\pm 0.01)}$	1.48 $^{(\pm 0.10)}$	1.17 $^{(\pm 0.01)}$
SwedishLeaf	4.08 $^{(\pm 0.09)}$	4.03 $^{(\pm 0.09)}$	4.01 $^{(\pm 0.08)}$	5.1 $^{(\pm 1.08)}$	4.1 $^{(\pm 0.15)}$
synthetic_control	21.79 $^{(\pm 0.19)}$	21.53 $^{(\pm 0.16)}$	21.31 $^{(\pm 0.16)}$	28.58 $^{(\pm 2.42)}$	21.72 $^{(\pm 0.16)}$
Trace	35.69 $^{(\pm 18.1)}$	28.75 $^{(\pm 19.7)}$	28.4 $^{(\pm 19.8)}$	72.47 $^{(\pm 33.6)}$	22.67 $^{(\pm 6.76)}$
Two_Patterns	13.69 $^{(\pm 1.46)}$	13.56 $^{(\pm 1.41)}$	13.55 $^{(\pm 1.41)}$	26.5 $^{(\pm 2.41)}$	17.67 $^{(\pm 2.64)}$
wafer	24.41 $^{(\pm 1.93)}$	23.71 $^{(\pm 1.97)}$	23.54 $^{(\pm 1.62)}$	55.56 $^{(\pm 6.33)}$	29.03 $^{(\pm 3.89)}$
yoga	11.93 $^{(\pm 0.36)}$	11.7 $^{(\pm 0.31)}$	11.63 $^{(\pm 0.32)}$	29.28 $^{(\pm 6.34)}$	12.64 $^{(\pm 1.41)}$

1. SSG-1 won in 98.3% of all trials against MM-1 with an average improvement of 29.1%. This result shows that SSG-1 substantially outperformed MM-1 and suggests to prefer SSG over MM in scenarios, where computation time matters.

2. After one iteration SSG-1 deviates merely 8.7% from the best solution found by SSG-50. In contrast MM-1 deviates 60.8% from the best solution found by MM-50. This indicates that SSG converges substantially faster than MM. We quantify this observation further in Section 5.2.3.

3. SSG-e won in 61.8% of all trials against MM-50 with an average improvement of 0.1%. Given the same amount of computation time defined by termination of MM, then SSG and MM perform comparable, though the results by SSG are slightly better than those obtained by MM.

4. SSG-50 won in 86.5% of all trials against MM-50 with an average improvement of 2.7%. This result indicates that the solutions found by SSG-50 further improved after termination of MM-50. Consequently, in situations where computation time is not an issue, SSG is more likely to return better solutions than MM.

5. One limitation of SSG is that its performance depends on the schedule with which the step size parameter is adapted. Finding an optimal schedule can take much longer than a run of DBA. To mitigate this limitation, we proposed a default schedule and applied it to all experiments. On the positive side, optimizing the step size schedule in a problem-dependent manner gives room for further improvement of the SSG algorithm. This advantage can be exploited in scenarios, where computation time is not so much a critical issue.

In summary, on average SSG finds solutions of similar quality faster than MM and is therefore better suited in situations, where sample size is large or many sample means need to be computed, such as in k-means for large datasets.

5.2. Comparison of Performance as a Function of the Sample Size

The goal of the second series of experiments is to assess the variation of the MM and SSG algorithms as a function of the sample size.

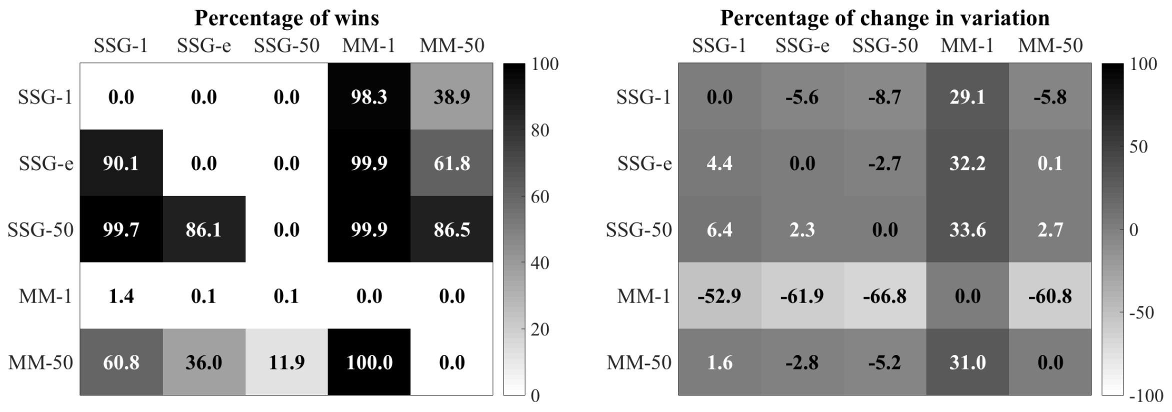


Figure 6: Percentage of wins (left) (row against column) and average percentage of change (right) over 720 trials.

5.2.1. Experimental Setup

The four largest datasets of the 24 UCR datasets shown in Table 1 were used: ECG5000, ElectricDevices, Two_Patterns, and wafer.

Given a dataset $\mathcal{X}$ and sample sizes $\mathcal{N} = \{N_1, \dots, N_r\}$, $N_i \leq |\mathcal{X}|$, which can be taken from Figure 7, an experiment was conducted as follows:

```
1.  given: dataset $\mathcal{X}$, sample sizes $\mathcal{N}$
2.  for 30 trials do
3.      for $N \in \mathcal{N}$ do
4.          randomly draw a subsample $\mathcal{X}_N$ of $\mathcal{X}$ of size $N$.
5.          randomly select initial solution $z \in \mathcal{X}_N$
6.          run SSG on $\mathcal{X}_N$ for 50 epochs with initial solution $z$
7.          run MM on $\mathcal{X}_N$ for 50 epochs with initial solution $z$
8.          record variations of MM and SSG after each epoch
9.      end for
10. end for
```

The step size schedule of SSG was chosen as in Section 5.1.

5.2.2. Results on Variation

Figure 7 shows the average variations and their standard deviations of the mean algorithms as a function of the sample size N. We made the following observations:

1. For large sample sizes ($N \geq 1000$), SSG-1 finds better solutions than MM-50 on average. Hence, for large sample sizes SSG finds better solutions in less time than MM on average.

2. The difference of average variations of SSG-50 and SSG-1 decreases with increasing sample size. For sample sizes $N \geq 500$ the solution qualities of SSG-1 and SSG-50 are comparable. This indicates that for sufficiently large sample sizes SSG may converge within the first epoch.

3. For increasing sample sizes, the difference of average variations of MM-1 and SSG-1 increases. Thus, increasing sample sizes have a positive influence on SSG-1 compared to MM-1. This indicates that the number of updates matters for solution quality. However, MM needs to cycle through the whole sample before making an update, whereas the number of updates made by SSG-1 increases with increasing sample size.

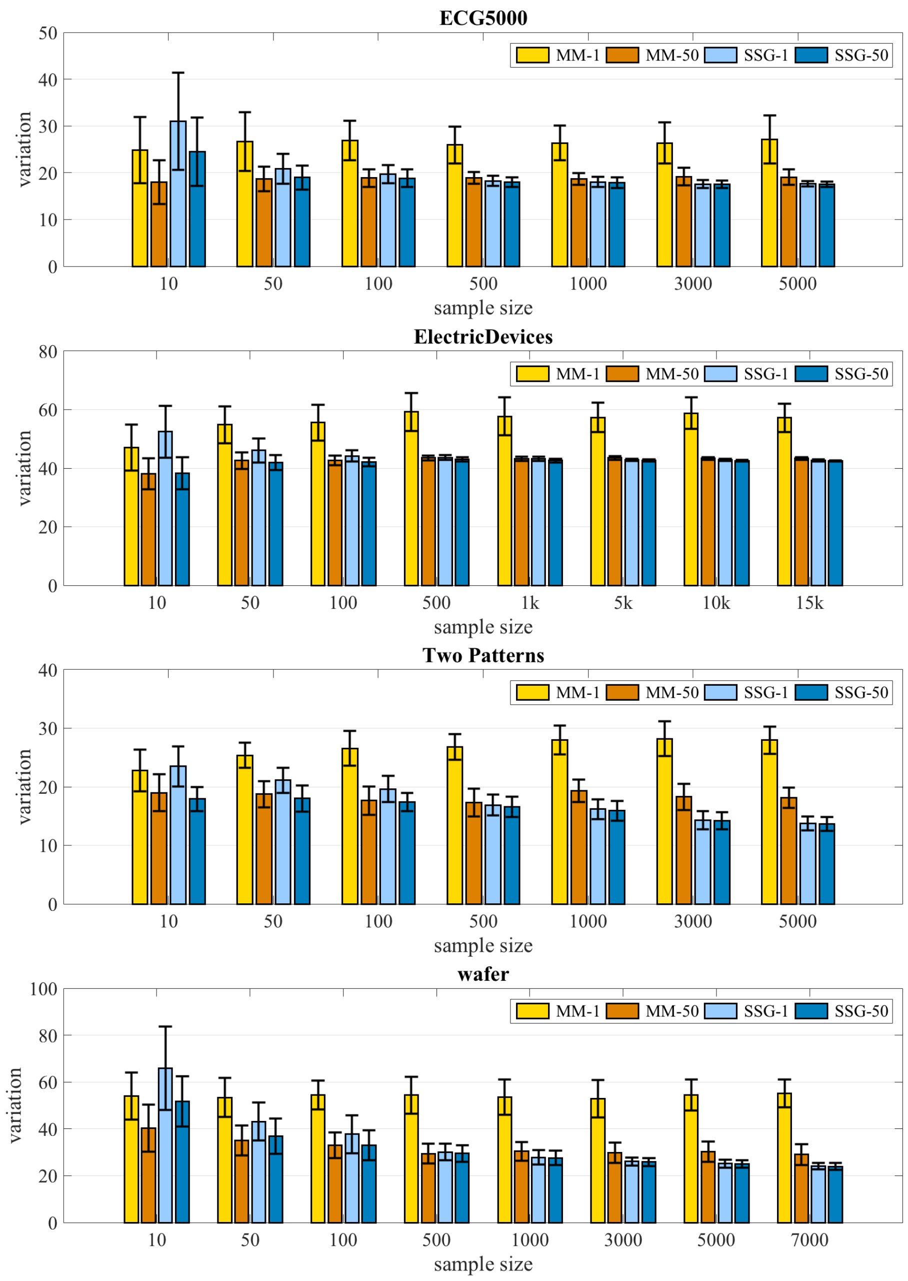


Figure 7: Average variation (lower is better) and standard deviation of MM-1, MM-50, SSG-1 and SSG-50.

4. For all but the smallest sample sizes SSG is more likely to find better solutions in shorter time than MM. For small sample sizes, MM finds better solutions in shorter time than SSG on average. These observations refine the findings of Section 5.1.

5. The standard deviations of SSG-1 and SSG-50 decrease faster with increasing sample size than the standard deviations of MM-1 and MM-50. This result shows that initialization of MM has a stronger effect on solution quality than the order with which SSG processes the sample time series. This finding disagrees with prior assumptions that the order of pairwise averaging causes inferior and unstable performance [24].

In summary, for all but the smallest sample sizes, SSG finds better solutions than MM on average. For

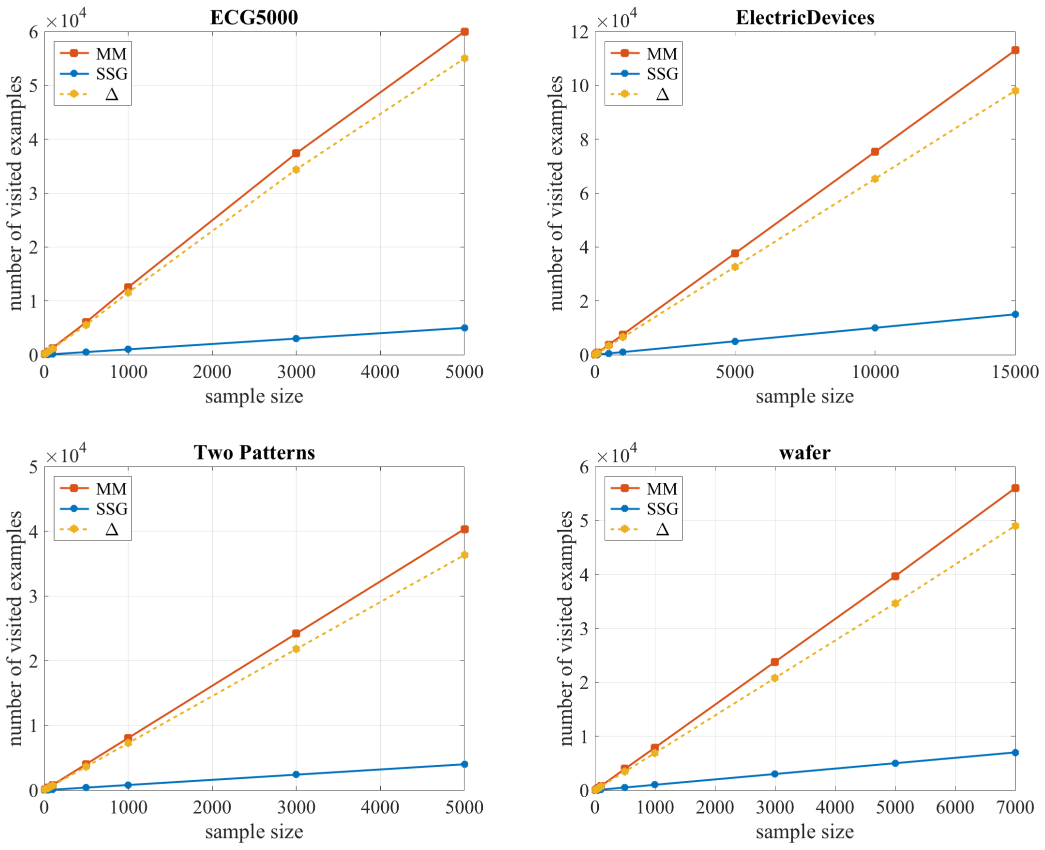


Figure 8: Average number of visited examples as a function of the sample size. The considered sample sizes can be inferred by the filled balls on the lines and correspond exactly to those shown in Figure 7. Due to the scaling of the x-axis, balls of small sample sizes overlap. MM refers to MM-50, SSG refers to SSG-e′, where e' is the number of epochs required by SSG to obtain at least the same solution quality as MM-50. Finally, Δ is the average difference of the number of visited examples between MM and SSG.

large sample sizes, performing a single epoch of SSG may be sufficient to find better and more robust solutions than MM-50. Moreover, for sufficiently large sample sizes SSG may converge within some small tolerance to a solution before it has processed a single epoch.

5.2.3. Results on Runtime

Next, we are interested in the runtimes of SSG and MM required to achieve a solution of approximately the same quality. The runtimes of SSG and MM are both $O(eNn^2)$, where e is the number of epochs until termination, N is the sample size, and n is the length of the time series. The quadratic factor n^2 is caused by computing the DTW distance between a sample time series and the current solution. Since complexity of both algorithms is identical, it is sufficient to measure the runtimes by counting the number of visited (processed) sample time series.

As solutions of approximately the same quality, we use the variations obtained by MM-50 as reference values. The number of visited examples by MM-50 until termination at epoch e is given by eN. Similarly, the number of visited examples by SSG-50 is $e'N$, where e' is the minimum number of epochs required by SSG-50 to achieve a solution of at least the same quality as MM-50. We refer to this variant of SSG as SSG-e′. To ensure solutions of approximately the same quality, we excluded all trials ($\sim 5.5\%$) for which SSG-50 was unable to find a solution of at least the same quality as MM-50.

Figure 8 summarizes the results. The results show that SSG-e′ found solutions of at least the same quality five up to ten times faster than MM-50. In addition, we observe that the runtimes of both algorithms increase linearly with the sample size, but with different speed. Linearity implies that the number of epochs is independent of the sample size, given that the samples are randomly drawn from the same distribution. This result confirms the observation from Section 5.2.2 that many updates based on single examples lead to faster convergence to a solution of similar quality than a few updates based on the entire sample. The important advantage of SSG over MM is that computation time per update does not grow with the sample size. These findings are largely in line with those reported in the machine learning and deep learning literature on batch and stochastic gradient descent [3].

5.3. Why Pairwise Averaging has Failed and how it can Succeed

The empirical and theoretical results provide new insight on incremental (pairwise averaging) methods for the sample mean problem in DTW spaces. Empirically, we have two results:

1. The symmetric-incremental methods NLAAF [11] and PSA [19] performed inferior than the asymmetric-batch MM algorithm [24, 32].

2. For non-small sample sizes, the asymmetric-incremental SSG algorithm performed superior than the asymmetric-batch MM algorithm.

The first empirical result swept away incremental methods and replaced them by research on batch methods. The second empirical result disagrees with the assumption that pairwise averaging is the cause of inferior and unstable performance compared to batch methods such as the MM algorithm [24]. Instead, we assume that two other factors cause the inferior performance of the incremental methods NLAAF and PSA:

1. Recall from Section 2.3 that symmetric-incremental methods result in increasingly long average time series. To adjust the length of the average time series, symmetric-incremental methods usually apply a post-processing step. This post-processing step ignores the necessary conditions of optimality. Thus it could happen that the post-processing step displaces the average far from a local minimum.

2. The way the step size is adapted could matter.

These considerations suggest to empirically reassess iterative methods such as PSA in a modified form: (i) replace symmetric averages by asymmetric averages; (ii) eventually consider a suitable schedule for adapting the step size; and (iii) optionally apply the MM algorithm after termination of the modified iterative method.

Note that the third option can be applied to any asymmetric-iterative method including the SSG algorithm. The role of the asymmetric-iterative method can be interpreted as a sophistic way to initialize MM. Conversely, the role of the MM algorithm can be interpreted as a post-processing step to ensure the necessary conditions of optimality.

6. Conclusion

This article provides a theoretical foundation for the sample mean problem in DTW spaces. The main contributions can be summarized as (1) novel directions for devising improved sample mean algorithms, (2) improved understanding of the sample mean problem, (3) improved understanding of existing mean algorithms.

Novel directions of mean algorithms are nonsmooth optimization methods and asymmetric-incremental methods. As a representative of nonsmooth optimization methods, we presented subgradient methods. As a representative of both, nonsmooth and asymmetric-incremental methods, we propose a stochastic subgradient (SSG) method. Experiments show that for increasing sample sizes the SSG algorithm is more stable and finds better solutions in shorter time than the DBA algorithm on average. Therefore, SSG is particularly useful if the sample size is non-small and in situations where several sample means have to be estimated several times such as in k-means clustering of large-scale temporal datasets. In addition, as a stochastic method, SSG can be applied in online settings, where merely one sample time series is available at a time.

Concerning contribution (2), we presented a mathematically sound way to explore analytical properties of the Fréchet function. This work answers the remaining of the four fundamental questions of an optimization problem (existence and uniqueness of a solution, formulation of necessary and sufficient conditions of optimality). In particular, we know the form of local minimizers and of a sample mean.

Regarding contribution (3), we showed that the DBA algorithm is a majorize-minimize algorithm and a special case of a subgradient method. In conclusion, we have a mathematically convenient formulation of DBA that simplifies analysis of the algorithm and directly leads to better suited implementations for programming languages that benefit from matrix calculus instead of loops. We used the reformulation of DBA to prove convergence to solutions satisfying the necessary conditions of optimality after a finite number of iterations and obtained a new termination criterion for DBA. Further, we showed that Soheily-Khah et al. [33] generalized DBA to weighted and kernelized DTW distances.

The theoretical and empirical results justify to further explore the two proposed directions – nonsmooth optimization and asymmetric-incremental methods – for devising sample mean algorithms. Methods belonging to the first direction include sophisticated nonsmooth optimization methods such as plane cutting, bundle, and gradient sampling techniques. The second direction includes asymmetric variants of the PSA and NLAAF algorithms that were originally introduced as symmetric-incremental methods.

Acknowledgements: B. Jain was funded by the DFG Sachbeihilfe `JA 2109/4-1`.

A. Theoretical Background and Proofs

This section provides the background and the proofs of the theoretical results stated in this article.

A.1. Time Warping Embeddings

Every warping path $p \in \mathcal{P}$ of length L induces two embeddings $\Phi, \Psi : \mathcal{T} \to \mathbb{R}^L$ such that the cost of aligning two time series x and y along warping path p can be expressed as

$$C_p(x,y) = \left\lVert \Phi(x) - \Psi(y) \right\rVert^2 . \tag{8}$$

To construct both embeddings, we assume that $e^k \in \mathbb{R}^n$ denotes the k-th standard basis vector with elements

$$e_i^k = \begin{cases} 1 & : \quad i = k \\ 0 & : \quad i \neq k \end{cases} .$$

Definition A.1. *Let $p = (p_1, \dots, p_L) \in \mathcal{P}$ be a warping path with points $p_l = (i_l, j_l)$. Then*

$$\Phi = \begin{bmatrix} e^{i_1} \\ \vdots \\ e^{i_L} \end{bmatrix} \in \mathbb{R}^{L \times n}, \quad \Psi = \begin{bmatrix} e^{j_1} \\ \vdots \\ e^{j_L} \end{bmatrix} \in \mathbb{R}^{L \times n},$$

is the pair of embedding matrices *induced by warping path p.*

The embedding matrices have full column rank n due to the boundary and step condition of the warping path. Thus, we can regard the embedding matrices of warping path p as injective linear maps $\Phi, \Psi : \mathbb{R}^n \to \mathbb{R}^L$ that embed time series x and y of length n into $\mathbb{R}^L$ by matrix multiplication Φx and Ψy. We show that Eq. (8) holds.

Proposition A.2. *Let Φ and Ψ be the embeddings induced by warping path $p \in \mathcal{P}$. Then*

$$C_p(x,y) = \left\lVert \Phi x - \Psi y \right\rVert^2 .$$

Proof. The assertion follows from

$$\left\lVert \Phi x - \Psi y \right\rVert^2 = \left\lVert \begin{bmatrix} x_{i_1} - y_{j_1} \\ \vdots \\ x_{i_L} - y_{j_L} \end{bmatrix} \right\rVert^2 = \sum_{l=1}^{L} \left(x_{i_l} - y_{j_l} \right)^2 = C_p(x,y).$$

□

The next result shows that we can express the warping and valence matrix as products of the embedding matrices.

Lemma A.3. *Let Φ and Ψ be the embeddings induced by warping path $p \in \mathcal{P}$. We have*

$$W = \Phi^{\mathsf{T}}\Psi$$
$$V = \Phi^{\mathsf{T}}\Phi,$$

where W and V are the warping and valence matrix of p.

Proof. Let $p = \big((i_1, j_1), \ldots, (i_L, j_L)\big)$.

1. We first show $W = \Phi^{\mathsf{T}}\Psi$. Observe that

$$[\Phi^{\mathsf{T}}\Psi]_{i,j} = \sum_{k=1}^{L} \Phi^{\mathsf{T}}_{i,k}\Psi_{k,j} = \sum_{k=1}^{L} \Phi_{k,i}\Psi_{k,j} = \sum_{k=1}^{L} e_i^{i_k} e_j^{j_k}.$$

Suppose that $(i,j) \in p$. Then there is a unique index $l \in [L]$ such that $(i,j) = (i_l, j_l)$, where uniqueness of l follows from the step condition of a warping path. Then we have

$$[\Phi^{\mathsf{T}}\Psi]_{i,j} = e_i^{i_l} e_j^{j_l} = e_i^{i} e_j^{j} = 1.$$

Now suppose that $(i,j) \notin p$. Then $(i,j) \neq (i_k, j_k)$ for all $k \in [L]$. Hence, we have $e_i^{i_k} e_k^{j_k} = 0$ for all $k \in [L]$ and therefore $[\Phi^{\mathsf{T}}\Psi]_{i,j} = 0$. Combining the results shows that the elements of the matrix $[\Phi^{\mathsf{T}}\Psi]$ are of the form

$$[\Phi^{\mathsf{T}}\Psi]_{i,j} = \left\{ \begin{array}{lcl} 1 & : & (i,j) \in p \\ 0 & : & \text{otherwise} \end{array} \right. ,$$

which precisely corresponds to the definition of a warping matrix W of path p.

2. We show $V = \Phi^{\mathsf{T}}\Phi$. Let $u \in \mathbb{R}^n$ denote the vector of all ones. By definition of the valence matrix V, the diagonal elements of $V_{i,i}$ are of the form

$$V_{i,i} = \sum_{j=1}^{n} W_{i,j} = \sum_{j=1}^{n} [\Phi^{\mathsf{T}}\Psi]_{i,j} = \sum_{j=1}^{n}\sum_{k=1}^{L} e_i^{i_k} e_j^{j_k} = \sum_{k=1}^{L} e_i^{i_k} \sum_{j=1}^{n} e_j^{j_k} = \left(\sum_{k=1}^{L} e_i^{i_k}\right)\left(u^{\mathsf{T}} e^{j_k}\right) = \sum_{k=1}^{L} e_i^{i_k}.$$

From $e_i^{i_k} e_i^{i_k} = e_i^{i_k}$ follows

$$V_{i,i} = \sum_{k=1}^{L} e_i^{i_k} e_i^{i_k} = \sum_{k=1}^{L} \Phi_{k,i}\Phi_{k,i} = \sum_{k=1}^{L} \Phi^{\mathsf{T}}_{i,k}\Phi_{k,i} = [\Phi^{\mathsf{T}}\Phi]_{i,i}$$

This shows the assertion for the diagonal elements of V. Since V is a diagonal matrix, it is sufficient to show that $[\Phi^{\mathsf{T}}\Phi]_{i,j} = 0$ for all distinct $i, j \in [n]$. Suppose that $i, j \in [n]$ with $i \neq j$. We have

$$[\Phi^{\mathsf{T}}\Phi]_{i,j} = \sum_{k=1}^{L} \Phi_{k,i}\Phi_{k,j},$$

where $\Phi_{k,i}$ and $\Phi_{k,j}$ is the i-th and j-th component of the standard basis vector e^{i_k}. Since $i \neq j$ by assumption, we find that $\Phi_{k,i}\Phi_{k,j} = 0$ for all $k \in [L]$. This shows that all off-diagonal elements of $[\Phi^{\mathsf{T}}\Phi]$ are zero. By combining the results we obtain the second assertion. □

A.2. Gradient of Component Functions: Proof of Prop. 3.2

Let $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ be a sample of N time series $x^{(k)} \in \mathcal{T}$. As shown in Section 3.1 the Fréchet function F of $\mathcal{X}$ is a pointwise minimum of the form

$$F(x) = \min_{\mathcal{C} \in \mathcal{P}^N} F_{\mathcal{C}}(x),$$

with component functions

$$F_{\mathcal{C}}(x) = \frac{1}{N}\sum_{k=1}^{N} C_{p^{(k)}}\left(x, x^{(k)}\right),$$

where $\mathcal{C} = \left(p^{(1)}, \ldots, p^{(N)}\right) \in \mathcal{P}^N$ is a configuration of warping paths associated with sample $\mathcal{X}$. We restate Prop. 3.2.

Proposition. *Let $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right) \in \mathcal{T}^N$ be a sample of time series and let $\mathcal{C} \in \mathcal{P}^N$ be a configuration of warping paths. The gradient of component function $F_\mathcal{C}$ at $x \in \mathcal{T}$ is of the form*

$$\nabla F_\mathcal{C}(x) = \frac{2}{N} \sum_{k=1}^N \left(V^{(k)}x - W^{(k)}x^{(k)}\right),$$

where $V^{(k)}$ and $W^{(k)}$ are the valence and warping matrix of warping path $p^{(k)} \in \mathcal{C}$ associated with time series $x^{(k)} \in \mathcal{X}$.

Proof. Let $\Phi^{(k)}$ and $\Psi^{(k)}$ denote the embeddings induced by warping path $p^{(k)} \in \mathcal{C}$ for all $k \in [N]$. From Prop. A.2 follows that

$$F_\mathcal{C}(x) = \frac{1}{N} \sum_{k=1}^N \left\| \Phi^{(k)}x - \Psi^{(k)}x^{(k)} \right\|^2.$$

The function $F_\mathcal{C}$ is differentiable with gradient

$$\begin{aligned}
\nabla F_\mathcal{C}(x) &= \frac{2}{N} \sum_{k=1}^N \Phi^{(k)\intercal} \left(\Phi^{(k)}x - \Psi^{(k)}x^{(k)}\right) \\
&= \frac{2}{N} \sum_{k=1}^N \left(\Phi^{(k)\intercal}\Phi^{(k)}x - \Phi^{(k)\intercal}\Psi^{(k)}x^{(k)}\right) \\
&= \frac{2}{N} \sum_{k=1}^N \left(V^{(k)}x - W^{(k)}x^{(k)}\right),
\end{aligned}$$

where the last line follows from Lemma A.3. □

A.3. Necessary Conditions of Optimality: Proof of Theorem 3.3

Theorem. *Let F be the Fréchet function of sample $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right) \in \mathcal{T}^N$. If $z \in \mathcal{T}$ is a local minimizer of F, then there is a configuration $\mathcal{C} \in \mathcal{P}^N$ such that the following conditions are satisfied:*

(C1) *$F(z) = F_\mathcal{C}(z)$.*

(C2) *We have*

$$z = \left(\sum_{k=1}^N V^{(k)}\right)^{-1} \left(\sum_{k=1}^N W^{(k)}\, x^{(k)}\right),$$

where $V^{(k)}$ and $W^{(k)}$ are the valence and warping matrix of $p^{(k)} \in \mathcal{C}$ for all $k \in [N]$.

Proof. **1.** We apply Prop. A.2, to write the Fréchet function F of $\mathcal{X}$ as

$$F(x) = \frac{1}{N} \sum_{k=1}^N \min_\mathcal{C} \left\| \Phi^{(k)}x - \Psi^{(k)}x^{(k)} \right\|^2,$$

where the minimum is taken over all configurations $\mathcal{C} \in \mathcal{P}^N$. The matrices $\Phi^{(k)}$ and $\Psi^{(k)}$ are the embeddings induced by the warping paths $p^{(k)} \in \mathcal{C}$. Suppose that $z \in \mathcal{T}$ is a local minimizer of F. Then there is a configuration $\mathcal{C}_* \in \mathcal{P}^N$ consisting of optimal warping paths $p_*^{(k)} \in \mathcal{P}_*\left(z, x^{(k)}\right)$ such that

$$F(z) = \frac{1}{N} \sum_{k=1}^N \left\| \Phi_*^{(k)}z - \Psi_*^{(k)}x^{(k)} \right\|^2 = F_{\mathcal{C}_*}(z),$$

where $\Phi_*^{(k)}$ and $\Psi_*^{(k)}$ are the embeddings induced by the optimal warping paths $p_*^{(k)} \in \mathcal{C}_*$. The last equation shows condition (C1).

2. The function

$$F_{\mathcal{C}_*}(x) = \sum_{k=1}^N \left\| \Phi_*^{(k)}x - \Psi_*^{(k)}x^{(k)} \right\|^2$$

is convex and differentiable. By Prop. 3.2, the gradient of $F_{\mathcal{C}_*}(x)$ is of the form

$$\nabla F_{\mathcal{C}_*}(x) = \frac{2}{N}\sum_{k=1}^{N}\left(V_*^{(k)}x - W_*^{(k)}x^{(k)}\right),$$

where $V_*^{(k)}$ and $W_*^{(k)}$ are the valence and warping matrix of $p_*^{(k)} \in \mathcal{C}_*$. Setting the gradient of $F_{\mathcal{C}_*}$ to zero and solving the equation gives

$$z_* = \left(\sum_{k=1}^{N} V_*^{(k)}\right)^{-1}\left(\sum_{k=1}^{N} W_*^{(k)}\, x^{(k)}\right) \tag{9}$$

as unique minimizer of $F_{\mathcal{C}_*}(x)$. Moreover, we have

$$F(z) = F_{\mathcal{C}_*}(z) \geq F_{\mathcal{C}_*}(z_*) \tag{10}$$
$$F(x) \leq F_{\mathcal{C}_*}(x) \tag{11}$$

for all $x \in \mathcal{T}$ by construction.

3. We distinguish between two cases:

1. $F_{\mathcal{C}_*}(z) = F_{\mathcal{C}_*}(z_*)$: This directly implies $z = z_*$, because $F_{\mathcal{C}_*}(x)$ has a unique minimizer. Then condition (C2) follows from Eq. (9).
2. $F_{\mathcal{C}_*}(z) > F_{\mathcal{C}_*}(z_*)$: We show that this case contradicts the assumption that z is a local minimizer. Let $\mathcal{B} = \mathcal{B}(z, \varepsilon)$ be the Euclidean ball with center z and radius $\varepsilon > 0$. Then for every $\varepsilon > 0$ there is a time series $x \in \mathcal{B}$ satisfying $F_{\mathcal{C}_*}(x) < F_{\mathcal{C}_*}(z)$ because $F_{\mathcal{C}_*}$ is convex and z is not a minimum of $F_{\mathcal{C}_*}$. From Eq. (10) and (11) follows that
$$F(x) \leq F_{\mathcal{C}_*}(x) < F_{\mathcal{C}_*}(z) = F(z).$$
Since ε can be arbitrarily small, we find that z is not a local minimizer, which contradicts our assumption. Hence, the first case applies. □

A.4. Sufficient Conditions of Optimality: Proof of Prop. 3.4

Proposition. *Let F be the Fréchet function of sample $\mathcal{X}$ and let $z \in \mathcal{T}$ be a time series. Suppose that there is a configuration $\mathcal{C} \in \mathcal{P}^N$ such that z satisfies the necessary conditions (C1) and (C2). If $\mathcal{C}$ is unique, then $F(z)$ is a local minimum.*

Proof. From uniqueness of $\mathcal{C}$ follows that the active set at z is of the form $\mathcal{A}_F(z) = \{F_{\mathcal{C}}\}$. Then $F(z) = F_{\mathcal{C}}(z)$ is continuously differentiable in a neighborhood of z with positive Hessian at z. Since z satisfies condition (C2), the gradient of $F_{\mathcal{C}}$ at z exists and vanishes. Then the assertion follows from the sufficient conditions of a local minimum of continuously differentiable functions. □

A.5. Convergence of the MM Algorithm: Proof of Theorem 4.1

We first introduce some notations and definitions. By $2^{\mathcal{X}}$ we denote the set of all subsets of some set $\mathcal{X}$. A *point-to-set map* on $\mathcal{X}$ is a map $\alpha : \mathcal{X} \to 2^{\mathcal{X}}$ that assigns each point $x \in \mathcal{X}$ to a subset $\alpha(x) \subseteq \mathcal{X}$. Let $\mathcal{Z}_* \subseteq \mathcal{X}$ be a solution set. A function $f : \mathcal{X} \to \mathbb{R}$ is a *descent function* for $\mathcal{Z}_*$ and α, if

(i) $x \notin \mathcal{Z}_*$, then $f(z) < f(x)$ for all $z \in \alpha(x)$

(ii) $x \in \mathcal{Z}_*$, then $f(z) \leq f(x)$ for all $z \in \alpha(x)$.

Suppose that $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ is a sample of N time series $x^{(k)} \in \mathcal{T}$. A configuration

$$\mathcal{C} = \left(p^{(1)}, \ldots, p^{(N)}\right) \in \mathcal{P}^N$$

is an *optimal configuration* for $z \in \mathcal{T}$, if $p^{(k)} \in \mathcal{P}_*\left(z, x^{(k)}\right)$ are optimal warping paths between z and $x^{(k)} \in \mathcal{X}$. By $\mathcal{O}(z) \subseteq \mathcal{P}^N$ we denote the set of optimal configurations for z. Note that $\mathcal{C} \in \mathcal{O}(z)$ yields $F(z) = F_{\mathcal{C}}(z)$ by definition.

We restate Theorem 4.1.

Theorem. *The MM algorithm terminates after a finite number of iterations at a solution satisfying the necessary conditions of optimality (C1) and (C2).*

Proof. The Fréchet function F can be written as

$$F(x) = \min_{\mathcal{C} \in \mathcal{P}^N} F_{\mathcal{C}}(x),$$

where $F_{\mathcal{C}}$ are the component functions of F. Then the point-to-set map α on $\mathcal{T}$ of the MM algorithm is of the form

$$\alpha(z) = \left\{ \operatorname*{argmin}_{x \in \mathcal{T}} F_{\mathcal{C}}(x) \; : \; \mathcal{C} \in \mathcal{O}(z) \right\}.$$

Let $\mathcal{Z}_*$ denote the solution set consisting of all time series that satisfy conditions (C1) and (C2). We show that the Fréchet function F is a descent function for $\mathcal{Z}_*$ and α. Suppose that $x \in \mathcal{T}$. Any element $z \in \alpha(x)$ is the unique minimum of a component function $F_{\mathcal{C}}$, where $\mathcal{C} \in \mathcal{O}(x)$ is an optimal configuration for x. We have

1. $F(z) \leq F_{\mathcal{C}}(z)$ by definition of the Fréchet function.
2. $F_{\mathcal{C}}(z) \leq F_{\mathcal{C}}(x)$, because z is a minimum of $F_{\mathcal{C}}$.
3. $F_{\mathcal{C}}(x) = F(x)$, because $\mathcal{C}$ is an optimal configuration.

Combining the three inequalities yields $F(z) \leq F(x)$. Suppose that $x \notin \mathcal{Z}_*$. Since z is the unique minimum of the convex function $F_{\mathcal{C}}$, we have $F_{\mathcal{C}}(z) < F_{\mathcal{C}}(x)$ giving $F(z) < F(x)$. This proves the properties (i) and (ii) of a descent function.

Let $z \in \mathcal{T}$ be the current solution. Then the next solution $z' \in \alpha(z)$ is the unique minimum of a component function $F_{\mathcal{C}}$, where $\mathcal{C} \in \mathcal{O}(z)$ is an optimal configuration for z. Suppose that the MM algorithm terminates at z'. We show that $z' \in \mathcal{Z}_*$. Observe that

$$F(z') \leq F_{\mathcal{C}}(z') = \min_{x} F_{\mathcal{C}}(x) \leq F(z).$$

Termination is enforced by $F(z') = F(z)$. In this case, we have $F(z') = F_{\mathcal{C}}(z')$, which is condition (C1). Condition (C2) holds by construction of the map α.

It remains to show that the MM algorithm terminates after a finite number of iterations. Let $\left(z^{(t)}\right)$ be a sequence with $z^{(t+1)} \in \alpha\left(z^{(t)}\right)$. Suppose that $\left(\mathcal{C}^{(t)}\right)$ is a sequence of optimal configurations $\mathcal{C}^{(t)} \in \mathcal{O}\left(z^{(t)}\right)$ for $z^{(t)}$ such that

$$z^{(t+1)} = \operatorname*{argmin}_{x \in \mathcal{T}} F_{\mathcal{C}^{(t)}}(x).$$

We assume that the sequence $\left(z^{(t)}\right)$ is infinite. Since F is a descent function and the MM algorithm does not terminate, we have $F\left(z^{(s)}\right) < F\left(z^{(t)}\right)$ for all $s, t \in \mathbb{N}$ such that $0 \leq t < s$. Observe that

$$F\left(z^{(s+1)}\right) \leq F_{\mathcal{C}^{(s)}}\left(z^{(s+1)}\right) \leq F\left(z^{(s)}\right) \leq F\left(z^{(t+1)}\right) \leq F_{\mathcal{C}^{(t)}}\left(z^{(t+1)}\right) \leq F\left(z^{(t)}\right).$$

Suppose that $\mathcal{C}^{(s)} = \mathcal{C}^{(t)}$. Since the minimum of component function $F_{\mathcal{C}^{(t)}}$ is unique, we find that $z^{(s+1)} = z^{(t+1)}$. This yields $F\left(z^{(s+1)}\right) = F\left(z^{(t+1)}\right)$, which contradicts our assumption that F is strictly decreasing. This shows that $\mathcal{C}^{(s)} \neq \mathcal{C}^{(t)}$ for all $s > t$. Hence, the configurations of the sequence $\left(\mathcal{C}^{(t)}\right)$ are mutually distinct. This contradicts our assumption that the sequence $\left(z^{(t)}\right)$ is infinite, because there are only finitely many different configurations. Consequently, the MM algorithm terminates after finitely many iterations to a solution satisfying the necessary conditions of optimality. □

B. Generalizations

In the main text, we considered the special case of univariate sample time series of fixed length. This section briefly sketches in two steps how to generalize the results of the univariate-fixed case. The first step relaxes the condition that the univariate sample time series are of fixed length. We call this case the univariate-variable case. The second step relaxes the condition that the time series are univariate, called the multivariate-variable case.

B.1. Generalization to the Univariate-Variable Case

The univariate-variable case assumes that $F : \mathcal{T}_n \to \mathbb{R}$ is a Fréchet function of sample $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ consisting of N univariate time series $x^{(k)} \in \mathcal{T}_{n_k}$ of possibly variable length $n_k \in \mathbb{N}$.

Suppose that $x \in \mathcal{T}_n$ is the argument of the Fréchet function and $y \in \mathcal{T}_m$ is a sample time series, that is $y = x^{(k)}$ and $m = n_k$ for some $k \in [N]$. The results carry over to the univariate-variable case for the following reasons:

1. The cost $C_p(x, y)$ of aligning x and y along warping path $p \in \mathcal{P}_{n,m}$ can be regarded as a differentiable convex function on $\mathbb{R}^n$. By adopting the same arguments as in Section 3, we find that the Fréchet function F of sample $\mathcal{X}$ is also a pointwise minimum of a set of differentiable convex functions and is therefore locally Lipschitz. Hence, the subdifferential calculus is also applicable to the univariate-variable case.

2. The embedding matrices $\Phi \in \mathbb{R}^{L\times n}$ and $\Psi \in \mathbb{R}^{L\times m}$ induced by a warping path $p \in \mathcal{P}_{n,m}$ of length L can be defined in a similar way as in Definition A.1. What has changed is that the rows of Ψ have the form of standard basis vectors from $\mathbb{R}^m$ rather than from $\mathbb{R}^n$.

3. The statement of Lemma A.3 and its proof carry over to the univariate-variable case. The warping matrix $W = \Phi^{\mathcal{T}}\Psi$ is a well-defined $(n \times m)$-matrix and the valence matrix $V = \Phi^{\mathcal{T}}\Phi$ is a well-defined $(n \times n)$-matrix. The proof of Lemma A.3 requires some noncritical adaptions to account for the length m of the sample time series x. Apart from this, the proof remains the same.

4. By construction, the expressions Vx and Wy are well-defined vectors in $\mathbb{R}^n$ regardless of the length m of sample time series y. Consequently, the expression

$$\sum_{k=1}^{N} V^{(k)}x - W^{(k)}x^{(k)}$$

is a well-defined vector in $\mathbb{R}^n$, where the $x^{(k)}$ are the sample time series of sample $\mathcal{X}$. The proofs of Prop. 3.2 and Theorem 3.3 directly carry over to the univariate-variable case. The gradient of F and condition (C2) of Theorem 3.3 are essentially of the same form as in the univariate-fixed case. What differs are the dimensions of the warping matrices.

5. The proofs of all other results are independent of the length of the sample time series.

B.2. Generalization to the Multivariate-Variable Case

The multivariate-variable case assumes that $F : \mathcal{T}_n \to \mathbb{R}$ is a Fréchet functions of sample $\mathcal{X} = \left(x^{(1)}, \ldots, x^{(N)}\right)$ consisting of N multivariate time series $x^{(k)} \in \mathcal{T}_{n_k}$ of possibly variable length $n_k \in \mathbb{N}$.

A d-dimensional time series x of length n can be represented as a matrix $x = (x_{i,j}) \in \mathbb{R}^{n\times d}$, where $d \geq 1$. Every column $x_{:j}$ of matrix x corresponds to a univariate *component time series* of the form

$$x_{:j} = (x_{1j}, \ldots, x_{nj}) \in \mathbb{R}^n$$

for all $j \in [d]$. Moreover, every row x_i of matrix x represents a d-dimensional feature vector

$$x_i = (x_{i1}, \ldots, x_{id}) \in \mathbb{R}^d.$$

for every time point $i \in [n]$.

We assume that all multivariate time series under considerations are of dimension d. Then by $\mathcal{T}_n$ we denote the set of multivariate time series of length n. Suppose that $x \in \mathcal{T}_n$ is the argument of the Fréchet function and $y \in \mathcal{T}_m$ is a sample time series, that is $y = x^{(k)}$ and $m = n_k$ for some $k \in [N]$. The results carry over to the multivariate-variable case for the following reasons:

1. Let $p = (p_1, \ldots, p_L) \in \mathcal{P}_{n,m}$ be a warping path between x and y with elements $p_l = (i_l, j_l)$ for all $l \in [L]$. The cost $C_p(x, y)$ of aligning x and y along warping path p is of the form

$$C_p(x, y) = \sum_{l=1}^{L} \left\|x_{i_l} - y_{j_l}\right\|^2.$$

By assuming that y is a fixed parameter, we can regard the cost C_p as a differentiable convex function on $\mathbb{R}^{n\times d}$. Then as in Section B.1, we find that the Fréchet function F of sample $\mathcal{X}$ is locally Lipschitz continuous as a pointwise minimum of a set of differentiable convex functions. Therefore, the subdifferential calculus is also applicable to the multivariate-variable case.

2. Next, we construct the embeddings of a warping path. Suppose that $p = (p_1, \dots, p_L) \in \mathcal{P}_{n,m}$ is a warping path between x and y. The multivariate time warping embeddings $\Phi' : \mathcal{T}_n \to \mathcal{T}_L$ and $\Psi' : \mathcal{T}_m \to \mathcal{T}_L$ induced by p are linear functions defined by

$$\Phi'(x) = \begin{bmatrix} x_{i_1} \\ \vdots \\ x_{i_L} \end{bmatrix}, \qquad \Psi'(y) = \begin{bmatrix} y_{j_1} \\ \vdots \\ y_{j_L} \end{bmatrix}.$$

We regard Φ' and Ψ' as linear mappings $\Phi' : \mathbb{R}^{n \times d} \to \mathbb{R}^{L \times d}$ and $\Psi' : \mathbb{R}^{m \times d} \to \mathbb{R}^{L \times d}$. Then the multivariate formulation of Prop. A.2 remains true with Frobenius norm $\|(x_{i,j})\|_F = \left(\sum_{i=1}^N \sum_{j=1}^d x_{i,j}^2\right)^{1/2}$:

$$C_p(x, y) = \left\|\Phi'(x) - \Psi'(y)\right\|_F^2.$$

3. The operations of Φ' and Ψ' induced by warping path $p \in \mathcal{P}_{n,m}$ can be expressed by componentwise matrix multiplications with the embedding matrices (cf. Section A.1) Φ and Ψ induced by p:

$$\Phi'(x) = \begin{bmatrix} \Phi x_{:1} & \dots & \Phi x_{:d} \end{bmatrix}, \qquad \Psi'(y) = \begin{bmatrix} \Psi y_{:1} & \dots & \Psi y_{:d} \end{bmatrix}.$$

Therefore, componentwise, everything reduces to the univariate-variable case. This is sufficient, because gradients are defined componentwise and the necessary conditions of optimality are derived from gradients. Hence, all other results carry over to the multivariate-variable case.

4. When considering the problem componentwise, it is important to note that the valence and warping matrices $V^{(k)}$ and $W^{(k)}$, resp., are induced by warping paths $p^{(k)} \in \mathcal{P}_{n,n_k}$ of a *single* configuration $\mathcal{C} = \left(p^{(1)}, \dots, p^{(N)}\right)$ and therefore identical for each component time series. For example the gradient $\nabla F_{\mathcal{C}}(x) \in \mathbb{R}^{n \times d}$ of a component function F_C at some $x \in \mathcal{T}_n$ is given by

$$\nabla F_{\mathcal{C}}(x) = \frac{2}{N} \sum_{k=1}^N \left[\left(V^{(k)} x_{:1} - W^{(k)} x_{:1}^{(k)}\right) \quad \dots \quad \left(V^{(k)} x_{:d} - W^{(k)} x_{:d}^{(k)}\right) \right].$$

References

References

[1] W.H. Abdulla, D. Chow, and G. Sin. Cross-words reference template for DTW-based speech recognition systems. *Conference on Convergent Technologies for Asia-Pacific Region*, 2003.

[2] A. Bagirov, N. Karmitsa, and M.M. Mäkelä. *Introduction to Nonsmooth Optimization.* Springer International Publishing, 2014.

[3] Y. Bengio. Practical Recommendations for Gradient-Based Training of Deep Architectures. *Neural Networks: Tricks of the Trade*, K.-R. Müller, G. Montavon, and G.B. Orr (eds.), Springer, 2013.

[4] J.F. Bonnans, J.C. Gilbert, C. Lemarechal, and C.A. Sagastizbal. *Numerical optimization: theoretical and practical aspects.* Springer Science+Business Media, 2006.

[5] Y. Chen, E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, and G. Batista. *The UCR Time Series Classification Archive.* URL: `www.cs.ucr.edu/~eamonn/time_series_data/`, 2015.

[6] F.H. Clarke. *Optimization and Nonsmooth Analysis.* Society for Industrial and Applied Mathematics, 1990.

[7] E. Dimitriadou, A. Weingessel, and K. Hornik. A combination scheme for fuzzy clustering. *International Journal of Pattern Recognition and Artificial Intelligence*, 16(07):901–912, 2002.

[8] Y. Ermoliev and V. Norkin. Stochastic generalized gradient method for nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 34(2):196–215, 1998.

[9] L.C. Evans and R.F. Gariepy. *Measure theory and fine properties of functions.* CRC Press, 1992.

[10] M. Fréchet. Les éléments aléatoires de nature quelconque dans un espace distancié. *Annales de l'institut Henri Poincaré*, 215–310, 1948.

[11] L. Gupta, D. Molfese, R. Tammana, and P.G. Simos. Nonlinear alignment and averaging for estimating the evoked potential. *IEEE Transactions on Biomedical Engineering*, 43(4):348–356, 1996.

[12] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology.* Cambridge University Press, 1997.

[13] V. Hautamaki, P. Nykanen, P. Franti. Time-series clustering by approximate prototypes. *International Conference on Pattern Recognition*, 2008.

[14] D. Hunter and K. Lange. A tutorial on MM algorithms. *The American Statistician* , 58(1):30-37, 2004.

[15] B.J. Jain. Statistical Analysis of Graphs. *Pattern Recognition*, 60:802–812, 2016.

[16] B.J. Jain and D. Schultz. A Reduction Theorem for the Sample Mean in Dynamic Time Warping Spaces. *arXiv preprint*, arXiv:1610.04460, 2016.

[17] R. Lummis. Speaker verification by computer using speech intensity for temporal registration. *IEEE Transactions on Audio and Electroacoustics*, 21(2):80–89, 1973.

[18] J.B. Kruskal and M. Liberman. The symmetric time-warping problem: From continuous to discrete. *Time warps, string edits and macromolecules: The theory and practice of sequence comparison*, pp. 125–161, 1983.

[19] V. Niennattrakul and C.A. Ratanamahatana. Shape averaging under time warping. *IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, 2009.

[20] V. Niennattrakul, D. Srisai, and C.A. Ratanamahatana. Shape-based template matching for time series data *Knowledge-Based Systems*, 26:1–8, 2012.

[21] V. Norkin. Stochastic generalized-differentiable functions in the problem of nonconvex nonsmooth stochastic optimization. *Cybernetics and Systems Analysis*, 22(6):804–809, 1986.

[22] T. Oates, L. Firoiu, and P.R. Cohen. Clustering Time Series with Hidden Markov Models and Dynamic Time Warping. IJCAI Workshop on Neural, Symbolic and Reinforcement Learning methods for Sequence Learning, 1999.

[23] S. Ongwattanakul and D. Srisai. Contrast enhanced dynamic time warping distance for time series shape averaging classification. *International Conference on Interaction Sciences: Information Technology, Culture and Human*, 2009.

[24] F. Petitjean, A. Ketterlin, and P. Gancarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition* 44(3):678–693, 2011.

[25] F. Petitjean and P. Gançarski. Summarizing a set of time series by averaging: From Steiner sequence to compact multiple alignment. *Theoretical Computer Science*, 414(1):76–91, 2012.

[26] F. Petitjean. Matlab and Java source code for DBA. doi:10.5281/zenodo.10432, 2014.

[27] F. Petitjean, G. Forestier, G.I. Webb, A.E. Nicholson, Y. Chen, and E. Keogh. Faster and more accurate classification of time series by exploiting a novel dynamic time warping averaging algorithm. *Knowledge and Information Systems*, 47(1):1–26, 2016.

[28] L.R. Rabiner and J.G. Wilpon. Considerations in applying clustering techniques to speaker-independent word recognition. *The Journal of the Acoustical Society of America*, 66(3): 663–673, 1979.

[29] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[30] D. Schultz and B.J. Jain. Sample Mean Algorithms for Averaging in Dynamic Time Warping Spaces. 10.5281/zenodo.216233, 2016.

[31] N.Z. Shor. *Minimization Methods for Non-Differentiable Functions.* Springer-Verlag Berlin, Heidelberg, 1985.

[32] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier. Progressive and Iterative Approaches for Time Series Averaging. *Workshop on Advanced Analytics and Learning on Temporal Data*, 2015.

[33] S. Soheily-Khah, A. Douzal-Chouakria, and E. Gaussier. Generalized k-means-based clustering for temporal data under weighted and kernel time warp. *Pattern Recognition Letters*, 75:63–69, 2016.

[34] P. Somervuo and T. Kohonen. Self-organizing maps and learning vector quantization for feature sequences. *Neural Processing Letters*, 10(2):151–159, 1999.

[35] D. Srisai and C.A. Ratanamahatana. Efficient time series classification under template matching using time warping alignment. *IEEE International Conference on Computer Sciences and Convergence Information Technology*, 2009.

[36] W. Zangwill. *Nonlinear Programming: A Unified Approach*, Prentice-Hall, 1969.