Time Series Decomposition Using the Fréchet Distance

Anne Driemel
University of Bonn, Germany

Jan Höckendorff
University of Cologne, Germany

Ioannis Psarros
Archimedes, Athena Research Center, Greece

Christian Sohler
University of Cologne, Germany

Abstract

In this paper, we introduce a new data analysis problem that aims to decompose a set of univariate time series into a small set of k base curves of length at most l such that the sum of Fréchet distances of the time series to a "Fréchet combination" of the base curves is minimized. Here, a Fréchet combination allows to combine individually scaled base curves using a k-dimensional traversal. We call the problem of finding a set of optimal base curves the Fréchet decomposition problem and we consider two variants: (a) the base curves can be arbitrary curves of bounded length and (b) the curves come from a given finite set of candidate curves. We think of the Fréchet decomposition problem as a Fréchet variant of principal component analysis. For the case of a single base curve we develop a $(1+\varepsilon)$-approximation algorithm for the Fréchet decomposition problem. Additionally we give an exact algorithm for the projection distance problem that asks to compute the distance of one given time series to a given set of k base curves. This allows us to design an exact algorithm for the Fréchet decomposition problem for general k when curves come from a fixed candidate set.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Time Series Analysis, Fréchet Distance, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ESA.2026.96

Related Version *Full Version*: `https://arxiv.org/abs/2607.04397` [20]

Supplementary Material
Software (Source Code): `https://github.com/jahoec/PCA-on-Curves` [18]
archived at `swh:1:dir:42897eeb29e82508265e89725746120e29ec7177`

Funding *Anne Driemel*: Affiliated with Lamarr Institute for Machine Learning and Artificial Intelligence; Supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.
Jan Höckendorff: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.
Ioannis Psarros: This work has been partially supported by project MIS 5154714 of the National Recovery and Resilience Plan Greece 2.0 funded by the European Union under the NextGenerationEU Program.
Christian Sohler: Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) – Project Number 459420781.

1 Introduction

A fundamental problem in exploratory data analysis and forecasting is to decompose a given signal into its (most important) components. Several methods for this problem are known, including principal component analysis (PCA; also known as Karhunen-Loewe transform or empirical orthogonal functions [31, 24, 43]), complex principal component analysis [26], independent component analysis [28], functional principal component analysis etc. One drawback of principal component analysis is that it cannot deal effectively with individual temporal distortion that may happen for a relevant pattern across the input time series. One reason why this may happen is that the signal is not sampled uniformly, but one could also think of physical effects that could lead to such a behaviour. For example, in the area of oceanography, one of the main factors in sea surface height analysis [39] is a change that corresponds to summer/winter, and so depending on the location on the planet, the underlying signal may be shifted. Some variants of PCA like the complex principal component analysis, are able to find temporal shifts of patterns in the input [26]. However, this comes at the expense of an intrinsic indeterminacy that complicates the interpretation of the results. Kernel PCA [32] can be viewed as addressing similar needs. There are kernels for variants of the Fréchet distance [38] and DTW [17, 16]. However, a fundamental problem with this approach is that the decomposition exists only in lifting to a high-dimensional feature space, and there is no reverse mapping from the feature space back to the input space. This is also called the preimage problem [25]. Otherwise classical techniques like spectral analysis [35], wavelet analysis [37] are used to determine patterns with respect to a single time series, which makes them unsuitable for our setting.

In the statistics literature, so-called function registration aims to find suitable warping functions for the input time series to optimize the fit to a template or reference function as a preprocessing step to a (functional) principal component analysis [42, 36]. This is usually done for each input time series individually by solving a regression problem and is sensitive to the choice of the template function. Sometimes pairwise alignments are used; refer to [41] for a recent survey. More recent approaches combine the fitting of alignments with a principal component analysis [29, 30]. However, these approaches typically lack convergence guarantees, not to mention a rigorous running time or complexity analysis.

This paper aims at developing new techniques to decompose a signal into components in a setting where the basic components that shape the signal can be stretched or compressed in time. The Fréchet distance is well-equipped to handle such distortion. Building upon the work on clustering of time series [21, 10, 14, 9, 34, 13, 12, 19], we are aiming at developing a signal decomposition technique that uses the Fréchet distance.

More concretely, assume we are given a set $X = \{x_1, \ldots, x_n\}$ of time series. For example, each could be a time-ordered sequence of measurements of the same time-varying signal at a specific location. We are looking for k base signals $b_1, \ldots, b_k$ that can explain all x_i simultaneously. In other words, we think of x_i as being generated by the k base signals and noise. One simple way to phrase this problem is to say the x_i can be generated by a linear combination of the b_i, and we have Gaussian error. In this case, one would try to find k base curves $b_1, \ldots, b_k$ that minimize

$$\sum_{i=1}^{n} \min_{(w_1,\ldots,w_k)} \|x_i - \sum_{j=1}^{k} w_j \cdot b_j\|^2.$$

The latter problem can be solved using principal component analysis.

In our case, even though we are measuring the same underlying process, we must also take into account the situation that the same signals appear in a different form (for example, they might be delayed or distorted). We therefore introduce a problem formulation that allows for distorted signals, based on the (discrete) Fréchet distance. The discrete Fréchet distance between two signals $x = (x_1, \ldots, x_n)$ and $y = (y_1, \ldots, y_m)$ is defined as the minimum over all aligned traversals of x and y (where one can either advance on x or y or both) of the maximum distance of the matched entries. It can be easily extended to a multidimensional traversal between an input time series x_i and k weighted base curves $w_i b_i$ where w_i denotes the weight of the i-th base curve b_i. Let B denote such a set of k base curves. We develop a notion of the span under the Fréchet distance, denoted with $\mathrm{span}_{\mathrm{F}}(B)$, which is a set that contains all time series that can be formed as the weighted sum of elements of B under combined traversals. Let $d_F(x_i, \mathrm{span}_{\mathrm{F}}(B))$ denote the standard Hausdorff-extension of the distance to the set $\mathrm{span}_{\mathrm{F}}(B)$. We can think of this as a distance to a projection onto this set. Then our goal will be to find a set of k base curves B that minimize

$$\sum_{i=1}^{n} d_F(x_i, \mathrm{span}_{\mathrm{F}}(B)).$$

We will refer to the problem of finding B as the Fréchet Decomposition Problem. More details will be given in Section 3.

1.1 Related work

There is extensive literature on the complexity of computing the Fréchet distance. Computing the distance for two curves of complexity m takes roughly $O(m^2)$ time both for the discrete and the continuous version [3, 2]. This is optimal under the strong exponential time hypothesis [6, 1], even for 1-dimensional curves [8, 11] and it holds even if we allow a (small) constant approximation.

Related to our problem is a variant of the discrete Fréchet distance, which optimizes over all possible translations [33, 27, 4, 7]. Here, the state of the art is by Bringmann et al. [7] who give a $\tilde{O}(m^{4.66\ldots})$ algorithm and complement this result with a lower bound of $m^{4-\varepsilon}$ conditioned on the strong exponential time hypothesis.

Closely related to our work is the problem of clustering time series data. Driemel et al. [21] defined the (k, l)-clustering problem for time series as follows: Given a set P of n time series of complexity at most m and parameters $k, l \in \mathbb{N}$ find k center time series of complexity at most l, such that (a) the maximum distance of an element in P to its closest center time series or (b) the sum of these distances is minimized. Variant (a) is referred to as (k, l)-center and (b) as (k, l)-median. Under the continuous Fréchet distance, they developed near linear time $(1 + \varepsilon)$-approximation algorithms for both clustering variants, assuming ε, k, and l are constants. They complement these algorithmic results with hardness results, showing that both (k, l)-median and (k, l)-center are NP-hard under continuous Fréchet distance. Approximating (k, l)-median for polygonal curves in arbitrary dimensions was recently studied by Buchin et al. [12]. Cheng and Huang give the first $(1 + \varepsilon)$-approximation algorithm for (k, l)-median under continuous Fréchet distance in $d > 1$ [14]. Both clustering problems are also NP-hard under the discrete Fréchet distance and even for the case $k = 1$ [9] [10]. Nath and Taylor [34] then developed a $(1+\varepsilon)$-approximation algorithm for (k, l)-median under discrete Fréchet distance, which runs in polynomial time. Driemel et al. recently developed a near-linear time approximation scheme for (k, l)-median clustering under discrete Fréchet, assuming that k and l are constant [19].

Buchin and Rohde [13] designed the first coreset construction for (k, l)-median under both variants of the Fréchet distance, where the size of the coreset has logarithmic dependence on the number of input curves. Recently, Cohen-Addad et al. introduced a new technique for coresets yielding an input-size–independent coreset for (k, l)-median [15].

1.2 Results & High Level Approach

The main conceptual contribution of this paper is the introduction of the Fréchet Decomposition Problem (Problem 1, Section 3) for which we obtain two main algorithmic results that hold for n time series of complexity at most m, approximation parameter $\varepsilon > 0$ and complexity l of the k base curves. In the following, we assume that l, k and ε are constants.

- a $(1+\varepsilon)$-approximation algorithm with $\tilde{O}(n^2 + nm)$ running time for the Fréchet Decomposition Problem for $k = 1$ (Corollary 21), and
- a $(1+\varepsilon)$-approximation algorithm with $\tilde{O}(nm + |C|^k n)$ running time for the Fréchet Decomposition Problem when the set of base curves is taken from a given finite set of candidate curves C for constant $k \geq 1$ (Corollary 25).

We consider the setting that $k = 1$ in Section 4. Here we give a constant-factor approximation algorithm with running time $\tilde{O}(n^2 m)$ for Problem 1 (Theorem 17) and extend this algorithm to a $(1+\varepsilon)$-approximation algorithm with running time in $\tilde{O}(nm^{4.5} + n^2 m)$ (Theorem 20). To achieve this result, we exploit that the problem definition allows us to control the infinity norms of the base curves and input time series, which is expressed by the key Lemmas 9 and 10. For the case $k = 1$, this allows us to show that either weight 1 or -1 is a constant approximation with respect to the optimal weight if the base curve and input time series are normalized appropriately. We then compute the minimum-error l-simplifications of the (appropriately normalized) input time series and choose the simplification that minimizes a simplified variant of our objective function where weights are restricted to $\{-1, 1\}$ to get a constant approximation for the Fréchet decomposition problem with $k = 1$. This is then further refined to a $(1+\varepsilon)$-approximation by discretizing the area around the vertices of the constant approximation with an appropriately chosen grid.

In Section 5, we give an exact algorithm for the Fréchet Projection Problem (Problem 2, Section 3) that computes the cost of a set of time series with respect to a given set of base curves. To do so, we start with a dynamic programming approach to solve a variant of Problem 2, where the weights are fixed. To extend this to the weighted setting, we construct an arrangement of hyperplanes such that for each weight vector in the same cell of the arrangement, it holds that the dynamic program takes the same sequence of decisions. Given a cell, we can then construct a linear program, based on the corresponding sequence of decisions in the dynamic program, to find the best weights in the given cell. By doing so for each cell in the arrangement, we obtain an algorithm that has running time $O(m^{2k+2.5})$ (Theorem 23). This algorithm then leads to an exact algorithm for the Fréchet Decomposition Problem in the setting where the set of base curves must come from a given finite set C. The running time of the algorithm is in $O(|C|^k nm^{2k+2.5})$ (Corollary 24).

We combine the results of both sections with a dimension reduction result of Driemel et al. (Theorem 2 in [19]) that allows us to improve the dependency on m in the running times.

2 Preliminaries

We represent a time series of length m as an m-dimensional vector $x \in \mathbb{R}^m$ over the reals and refer to m as the *complexity* of the time series.

In the following, we will introduce the discrete Fréchet distance. To do so we start with the standard definition that uses the notion of traversals.

▶ **Definition 1** (traversal). *Given two time series $x = (x_1, \dots, x_m) \in \mathbb{R}^m$ and $y = (y_1, \dots, y_l) \in \mathbb{R}^l$ a* traversal *T is a sequence of index pairs $(i, j) \in [m] \times [l]$ satisfying:*

1. *T starts with pair $(1, 1)$*

2. *T ends with pair (m,l)*
3. *if (i,j) appears in T, it can only be followed by either $(i,j+1),(i+1,j)$ or $(i+1,j+1)$.*

If a pair (i,j) appears in T, we say that x_i and y_j are matched *in traversal T.*

Next, we state an alternative notation that uses matrices to model the traversals, which was introduced in [19]. Here, a traversal is encoded by special matrices that map an l-dimensional vector x to a t-dimensional space by copying some of the entries of v while keeping their relative order. These traversal matrices are defined as follows.

▶ **Definition 2** (traversal matrices). *$M \in \{0,1\}^{t\times l}$ is a traversal matrix if it satisfies:*
1. *The rows of M are standard unit vectors,*
2. *$M(1,1)=1$,*
3. *$M(t,l)=1$, and*
4. *For rows $i\in[t-1]$ and columns $j\in[l-1]$ it holds that if $M(i,j)=1$ then either $M(i+1,j+1)=1$ or $M(i+1,j)=1$. If $i\in[t-1]$ and $j=\ell$ then $M(i+1,j)=1$.*

For $t,l\in\mathbb{N}$ we denote with $\mathcal{M}_l^t$ the set of all traversal matrices of dimensions (t,l). A pair of traversal matrices (M_m,M_l) has *matching dimension* for two time series $x\in\mathbb{R}^m$ and $y\in\mathbb{R}^l$, if there is $t\in\mathbb{N}$ such that $t\geq\max(m,l)$, $M_m\in\mathcal{M}_m^t$, and $M_l\in\mathcal{M}_l^t$. When the dimensions t and l are clear from the context, then we will not mention $\mathcal{M}_l^t$ explicitly. Note that one can represent any traversal of $x=(x_1,\ldots,x_m)$ and $y=(y_1,\ldots,y_l)$ by a pair of traversal matrices (M_m,M_l) of matching dimension. Additionally the t-dimensional vectors $M_m\cdot(1,2,\ldots,m)^T$ and $M_l\cdot(1,2,\ldots,\ell)^T$ denote the index of the entry of x and y that is mapped to the corresponding entry in the t-dimensional vectors M_mx and M_ly, respectively. This allows us to identify (M_m,M_l) with a sequence T' of index pairs whose r-th entry is the pair (i,j) where i is the r-th entry of $M_m\cdot(1,2,\ldots,m)^T$ and j is the r-th entry of $M_l\cdot(1,2,\ldots,l)$. One obtains a traversal by removing multiple neighboring occurrences of pairs (i,j) from the sequence T'. For a traversal matrix M of dimensions (t,m), we will call the vector $M\cdot(1,2,\ldots,m)^T$ the *traversal identifier* of M.

▶ **Definition 3** (Discrete Fréchet distance). *Given time series $x\in\mathbb{R}^m, y\in\mathbb{R}^l$ the* discrete Fréchet distance *$d_F(x,y)$ between x and y is defined as*

$$d_F(x,y)=\min\|M_mx-M_ly\|_\infty\,,$$

where the minimum is over all pairs of traversal matrices M_m,M_l with matching dimension. As a standard extension we define for any set $Y\subseteq\bigcup_{l\in\mathbb{N}}\mathbb{R}^l$

$$d_F(x,Y)=\min_{y\in Y}d_F(x,y).$$

We introduce the notion of the Fréchet span of a set of time series $Y=\{y_1,\ldots,y_k\}$. The span consists of all summations of time series that are formed from $y_1,\ldots,y_k$ by applying traversal matrices of matching dimension to each y_i.

▶ **Definition 4** (Fréchet closure). *Let $y\in\mathbb{R}^l$, the Fréchet t-closure of y is defined as*

$$\mathrm{closure}_t(y)=\left\{My\mid M\in\mathcal{M}_l^t\right\}.$$

It is the set of all Fréchet expansions to vectors of length t. The closure is the union of the t-closures over all $t\in\mathbb{N}$.

▶ **Observation 5.** *The discrete Fréchet distance between x and y can be written as the minimum ℓ_∞ distance between the two closures of x and y:*

$$d_F(x,y) = \min_{t\in\mathbb{N}} \{ \ \|\hat{x}-\hat{y}\|_\infty \mid \hat{x} \in \mathrm{closure}_t(x), \hat{y} \in \mathrm{closure}_t(y)\}$$

▶ **Definition 6** (Fréchet span)**.** *For $Y = \{y_1,\dots,y_k\} \subset \mathbb{R}^l$, we define the Fréchet span of Y as the set of linear combinations of the elements of the closures of $y_i \in Y$.*

$$\mathrm{span_F}(Y) = \bigcup_{w\in\mathbb{R}^k} \bigcup_{t\in\mathbb{N}} \left\{ \sum_{i=1}^{k} w_i x_i \mid x_i \in \mathrm{closure}_t(y_i) \right\}$$

Later in the paper, we will use the concept of minimum-error l-simplification, which can be computed in $O(lm\log(m)\log(m/l))$ time as in [5] and is defined as follows.

▶ **Definition 7** (minimum-error l-simplification)**.** *A time series $x' \in \mathbb{R}^l$ is a minimum-error l-simplification of a time series $x \in \mathbb{R}^m$, if for any time series $y \in \mathbb{R}^l$, $d_F(x,x') \le d_F(x,y)$.*

We will further utilize the following theorem of Driemel et al. [19] to reduce the complexity of the input time series to improve the running times of our results.

▶ **Theorem 8** ([19])**.** *Let $\varepsilon \in (0,1)$ and $l \in \mathbb{N}$ be constants. There exists a constant $d_0 = d_0(\varepsilon, l)$ such that for arbitrary $m \in \mathbb{N}$ and input time series $x \in \mathbb{R}^m$ one can compute in time $O(m\log^2 m)$ a time series $z \in \mathbb{R}^{d_0}$ such that for all $y \in \mathbb{R}^l$,*

$$(1-\varepsilon)d_F(x,y) \le d_F(z,y) \le (1+\varepsilon)d_F(x,y).$$

Hyperplane arrangements. A *hyperplane* is a set $\{x \in \mathbb{R}^d : \langle a,x\rangle = b\}$ where $a \in \mathbb{R}^d \setminus \{0\}$ and $b \in \mathbb{R}$. A *half-space* is a set $\{x \in \mathbb{R}^d : \langle a,x\rangle \le b\}$ where $a \in \mathbb{R}^d \setminus \{0\}$ and $b \in \mathbb{R}$, and an *open half-space* is a set $\{x \in \mathbb{R}^d : \langle a,x\rangle < b\}$ where $a \in \mathbb{R}^d \setminus \{0\}$ and $b \in \mathbb{R}$. An *arrangement* $\mathcal{A}(H)$ of a finite set H of hyperplanes in $\mathbb{R}^d$ is a partition of $\mathbb{R}^d$ into cells: a d-dimensional cell in $\mathcal{A}(H)$ is a maximal connected region of $\mathbb{R}^d$ not intersected by any hyperplane in H. Unless otherwise stated, we refer to d-dimensional cells as cells. A cell F of $\mathcal{A}(H)$ can be described by a sign vector in $\{-1,+1\}^{|H|}$ which is defined with respect to a fixed orientation of each hyperplane: each $h \in H$ partitions $\mathbb{R}^d$ into three regions: h, and the two open half-spaces determined by it; one of them is arbitrarily assigned to $+1$ and the other one is assigned to -1. For a cell F of an arrangement $\mathcal{A}(H)$, $\mathrm{cl}(F)$ denotes its closure.

3 Problem statement

Our goal is to decompose our set of input time series into base curves (plus error). For a set of base curves $B = \{b_1,\dots,b_k\} \subset \mathbb{R}^l$, the error associated with an input curve x is equal to the distance between x and the Fréchet span of B, $\mathrm{span_F}(B)$. Our goal is to minimize the sum of such errors over all input curves. This is formalized as follows.

▶ **Problem 1** (Fréchet Decomposition)**.** *Let $k, l \ge 1$ be integers. Given a set $P \subset \mathbb{R}^m$ of n time series, find a set $B \subset \mathbb{R}^l$ of k base curves that minimize*

$$\mathrm{cost}(P,B) = \sum_{x\in P} d_F(x, \mathrm{span_F}(B)).$$

A closely related, but likely simpler problem asks for the "projection" of a given input time series x to a fixed set of base curves.

▶ **Problem 2** (Fréchet Projection Distance)**.** *Let $x \in \mathbb{R}^m$ be a time series and $B = \{b_1,\dots,b_k\}$ be a set of time series from $\mathbb{R}^l$, where k and l are fixed integers. The projection distance problem is to compute $d_F(x, \mathrm{span_F}(B))$.*

4 $(1+\varepsilon)$-approximation for finding a single base curve $(k=1)$

In this section, we present an algorithm achieving a $(1+\varepsilon)$-approximation for the Fréchet Decomposition problem with $k=1$. Our approach begins by computing a constant approximation, which is subsequently refined to obtain a $(1+\varepsilon)$-approximation. Missing proofs of this section can be found in the corresponding section of the full version of the paper [20].

We begin our exposition with the algorithm that computes a constant approximation. Algorithm 1 first computes minimum-error l-simplifications of the (appropriately normalized) input time series and chooses the simplification that minimizes a simplified variant of our objective function where weights are restricted to $\{-1,1\}$. The first goal of this section is to prove correctness and bound the running time of Algorithm 1.

Algorithm 1 Constant-approximation algorithm for finding a single base curve.

procedure `ConstantApprox` $(P \subset \mathbb{R}^m,\ l \in \mathbb{N})$
    for each $x \in P$ **do**
        $y_x \leftarrow$ minimum-error l-simplification of $\frac{x}{\|x\|_\infty}$
    $min \leftarrow \infty$
    for each $x \in P$ **do**
        $c \leftarrow \sum_{z\in P} \|z\|_\infty \cdot \min_{w\in\{-1,1\}} d_F(\frac{z}{\|z\|_\infty}, w \cdot y_x)$
        if $min > c$ **then**
            $min \leftarrow c$
            $y \leftarrow y_x$
    return y

We begin with two lemmas that show the effect on the cost after either scaling the base curves or the input time series. Note that while these are stated for general k, we will only use them for $k=1$.

Lemma 9. *Given time series $x \in \mathbb{R}^m$ and $B = \{b_1, \ldots, b_k\} \subset \mathbb{R}^l$ with $l \le m$. Let $\beta \in (\mathbb{R} \setminus \{0\})^k$ and $B' = \{\beta_1 b_1, \ldots, \beta_k b_k\}$ then it holds that*

$$d_F(x, \mathrm{span_F}(B)) = d_F(x, \mathrm{span_F}(B')).$$

Proof.

$$\begin{aligned}
d_F(x, \mathrm{span_F}(B)) &= \min_{\substack{w_1,\ldots,w_k\in\mathbb{R}\\ M_0,M_1,\ldots,M_k}} \|M_0 \cdot x - \sum_{j=1}^{k} w_j \cdot M_j \cdot b_j\|_\infty \\
&= \min_{\substack{w_1,\ldots,w_k\in\mathbb{R}\\ M_0,M_1,\ldots,M_k}} \|M_0 \cdot x - \sum_{j=1}^{k} w_j \cdot M_j \cdot \frac{\beta_j}{\beta_j} \cdot b_j\|_\infty \\
&= \min_{\substack{w_1,\ldots,w_k\in\mathbb{R}\\ M_0,M_1,\ldots,M_k}} \|M_0 \cdot x - \sum_{j=1}^{k} \frac{w_j}{\beta_j} \cdot M_j \cdot \beta_j \cdot b_j\|_\infty \\
&= \min_{\substack{w'_1,\ldots,w'_k\in\mathbb{R}\\ M_0,M_1,\ldots,M_k}} \|M_0 \cdot x - \sum_{j=1}^{k} w'_j \cdot M_j \cdot \beta_j \cdot b_j\|_\infty \\
&= d_F(x, \mathrm{span_F}(B'))
\end{aligned}$$

◀

▶ **Lemma 10.** *Given time series $x \in \mathbb{R}^m$ and $B = \{b_1, \dots, b_k\} \subset \mathbb{R}^l$ with $l \leq m$. For any $\alpha \in \mathbb{R} \setminus \{0\}$ it holds that*

$$d_F(x, \mathrm{span_F}(B)) = \frac{1}{|\alpha|} d_F(\alpha \cdot x, \mathrm{span_F}(B)).$$

Proof.

$$\begin{aligned}
d_F(x, \mathrm{span_F}(B)) &= \min_{\substack{w_1,\dots,w_k \in \mathbb{R}\\ M_0, M_1, \dots, M_k}} \|M_0 \cdot x - \sum_{j=1}^{k} w_j \cdot M_j \cdot b_j\|_\infty \\
&= \min_{\substack{w_1,\dots,w_k \in \mathbb{R}\\ M_0, M_1, \dots, M_k}} \|\frac{\alpha}{\alpha} \cdot M_0 \cdot x - \sum_{j=1}^{k} w_j \cdot M_j \cdot \frac{\alpha}{\alpha} \cdot b_j\|_\infty \\
&= \frac{1}{|\alpha|} \min_{\substack{w_1,\dots,w_k \in \mathbb{R}\\ M_0, M_1, \dots, M_k}} \|\alpha \cdot M_0 \cdot x - \sum_{j=1}^{k} w_j \cdot \alpha \cdot M_j \cdot b_j\|_\infty \\
&= \frac{1}{|\alpha|} \min_{\substack{w'_1,\dots,w'_k \in \mathbb{R}\\ M_0, M_1, \dots, M_k}} \|\alpha \cdot M_0 \cdot x - \sum_{j=1}^{k} w'_j \cdot M_j \cdot b_j\|_\infty \\
&= \frac{1}{|\alpha|} d_F(\alpha \cdot x, \mathrm{span_F}(B))
\end{aligned}$$ ◀

We will utilize Lemma 9 and Lemma 10 to assume without loss of generality that all input time series and base curves have their infinity norm realized by a non-negative value, since we can always multiply them by -1 without changing the cost. Additionally, by Lemma 9, we will assume in the following that the base curves have ℓ_∞ norm 1.

Next, we make an observation on lower and upper bounds of the discrete Fréchet distance by identifying the value pairs that have the smallest and largest differences. In the following, for $v = (v_1, \dots, v_r) \in \mathbb{R}^r$ we will write $\max(v) = \max_{1 \leq i \leq r} v_i$ and $\min(v) = \min_{1 \leq i \leq r} v_i$.

▶ **Observation 11.** *For time series $x \in \mathbb{R}^m$, $b \in \mathbb{R}^l$ and $w \in \mathbb{R}$ it holds that*
- $d_F(x, wb) \leq \max\{|\max(x) - \min(wb)|, |\min(x) - \max(wb)|\}$*, and*
- $d_F(x, wb) \geq \max\{|\max(x) - \max(wb)|, |\min(x) - \min(wb)|\}$.

Note that since a time series v gets flipped by multiplying it with a negative weight $w \in \mathbb{R}_{<0}$, it holds $w \max(v) = \min(wv)$ and $w \min(v) = \max(wv)$.

To show that Algorithm 1 is correct, we first consider the problem of finding the weight that minimizes $d_F(x, w \cdot b)$ for fixed $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^l$. Note that if $\|b\|_\infty = 0$, any weight minimizes $d_F(x, w \cdot b)$ regardless of x. On the other hand, if $\|x\|_\infty = 0$ then weight 0 minimizes $d_F(x, w \cdot b)$, since x has to be all zeroes. We will therefore only consider time series x and b with $\|x\|_\infty, \|b\|_\infty \neq 0$.

Now we show that any optimal weight is contained in an interval with its boundaries depending on $\|x\|_\infty$ and $\|b\|_\infty$.

▶ **Lemma 12.** *For time series $x \in \mathbb{R}^m$ with $\|x\|_\infty = \max(x)$ and $b \in \mathbb{R}^l$ with $\|b\|_\infty = \max(b)$ it holds that*

$$\arg\min_{w \in \mathbb{R}} d_F(x, w \cdot b) \subseteq \left[\frac{-2\|x\|_\infty}{\|b\|_\infty}, \frac{2\|x\|_\infty}{\|b\|_\infty}\right].$$

Next, we show that the weight $\frac{\|x\|_\infty}{\|b\|_\infty}$ gives a constant approximation for the best non-negative weight.

▶ **Lemma 13.** *For time series $x \in \mathbb{R}^m$ with $\|x\|_\infty = \max(x)$ and $b \in \mathbb{R}^l$ with $\|b\|_\infty = \max(b)$ it holds that*

$$d_F\left(x, \frac{\|x\|_\infty}{\|b\|_\infty} \cdot b\right) \le 2 \min_{w \in \mathbb{R}_{\ge 0}} d_F(x, w \cdot b).$$

Next, we investigate non-positive weights. Here, we first show that an additive error can occur when a positive weight is used instead of a corresponding negative weight.

▶ **Lemma 14.** *Given time series $x \in \mathbb{R}^m$, $b \in \mathbb{R}^l$, and a non-positive weight $w \in \mathbb{R}_{\le 0}$, it holds that*

$$d_F(x, |w| \cdot b) \le d_F(x, w \cdot b) + 2 \cdot |w| \cdot \|b\|_\infty.$$

We can now show that using either -1 or 1 as weight is a constant approximation for any weight if x and b are scaled in a certain way through Lemmas 9 and 10.

▶ **Lemma 15.** *Given time series $x \in \mathbb{R}^m$ and $b \in \mathbb{R}^l$ such that $\|x\|_\infty = \|b\|_\infty = \max(x) = \max(b)$, it holds that*

$$\min_{w \in \{-1,1\}} d_F(x, w \cdot b) \le 10 \min_{w \in \mathbb{R}} d_F(x, w \cdot b).$$

By scaling the input time series to satisfy the condition of Lemma 15 we have to deal with weighted distances. The next lemma shows that we can find a constant factor approximation among the simplifications of the scaled input time series with respect to a modified variant of the objective function.

▶ **Lemma 16.** *Given a set of time series $\{x_1, \dots, x_n\} \subset \mathbb{R}^m$, parameters $a_1, \dots, a_n \in \mathbb{R} \setminus \{0\}$, and $l \in \mathbb{N}$, let $b' \in \arg\min_{b \in \mathbb{R}^l} \sum_{i=1}^n |a_i| \min_{w \in \{-1,1\}} d_F\left(\frac{x_i}{a_i}, w \cdot b\right)$ and y_j be a minimum-error l-simplification of $\frac{x_j}{a_j}$, where $j \in \arg\min_{i \in [n]} \min_{w \in \{-1,1\}} d_F\left(\frac{x_i}{a_i}, w \cdot b'\right)$. Then,*

$$\sum_{i=1}^n |a_i| \min_{w \in \{-1,1\}} d_F\left(\frac{x_i}{a_i}, w \cdot y_j\right) \le 3 \sum_{i=1}^n |a_i| \min_{w \in \{-1,1\}} d_F\left(\frac{x_i}{a_i}, w \cdot b'\right).$$

We are now ready to analyze Algorithm 1. We use the fact that Algorithm 1 normalizes all the input time series to the same ℓ_∞ norm, which allows us to prove the following theorem by combining Lemma 15 and Lemma 16.

▶ **Theorem 17.** *Given n time series $P \subset \mathbb{R}^m$ and $l \in \mathbb{N}$ then Algorithm 1 computes in time $\tilde{O}(n^2 ml)$ a 30-approximation for the Fréchet decomposition problem with $k = 1$.*

Proof. Let $b^* \in \arg\min_{b \in \mathbb{R}^l} \sum_{x \in P} \min_{w \in \mathbb{R}} d_F(x, w \cdot b)$ be an optimal solution for the Fréchet decomposition problem with $k = 1$. By Lemma 9 we can assume that $\|b^*\|_\infty = \max(b^*) = 1$. Let y be the time series returned by the algorithm. We show that y is a 30-approximation as follows. By Lemma 10, we obtain

$$\sum_{x \in P} \min_{w \in \mathbb{R}} d_F(x, w \cdot y) = \sum_{x \in P} \|x\|_\infty \min_{w \in \mathbb{R}} d_F(\frac{x}{\|x\|_\infty}, w \cdot y) \le \sum_{x \in P} \|x\|_\infty \min_{w \in \{-1,1\}} d_F(\frac{x}{\|x\|_\infty}, w \cdot y).$$

Next, by Lemma 16,

$$\begin{aligned}\sum_{x \in P} \min_{w \in \mathbb{R}} d_F(x, w \cdot y) &\le 3 \min_{b \in \mathbb{R}^l} \sum_{x \in P} \|x\|_\infty \min_{w \in \{-1,1\}} d_F(\frac{x}{\|x\|_\infty}, w \cdot b) \\ &\le 3 \sum_{x \in P} \|x\|_\infty \min_{w \in \{-1,1\}} d_F(\frac{x}{\|x\|_\infty}, w \cdot b^*),\end{aligned}$$

and, by Lemma 15 and Lemma 10,

$$\sum_{x\in P} \min_{w\in\mathbb{R}} d_F(x, w\cdot y) \le 30 \sum_{x\in P} \|x\|_\infty \min_{w\in\mathbb{R}} d_F\Big(\frac{x}{\|x\|_\infty}, w\cdot b^*\Big) = 30 \sum_{x\in P} \min_{w\in\mathbb{R}} d_F(x, w\cdot b^*).$$

The minimum-error l-simplifications of the time series can be computed in time $\tilde{O}(nlm)$ [5]. Finding the best time series among the simplifications takes $O(n^2ml)$ time. ◀

Our next step is to show that we can further improve the quality of the solution to an approximation ratio of $1+\varepsilon$. We show now that for an optimal base curve b^*, any time series inducing a small cost is either near b^* or near $-b^*$.

▶ **Lemma 18.** *Given a set of n time series $P \subset \mathbb{R}^m$, let $b^* \in \mathbb{R}^l$ be an optimal base curve and $b' \in \mathbb{R}^l$ be a time series with cost Δ'. Then,*

$$\min_{w\in\{-1,1\}} d_F(b', w\cdot b^*) \le \frac{40\Delta'}{\sum_{x\in P}\|x\|_\infty}.$$

Now, given a time series $b' \in \mathbb{R}^l$ which is either near to b^* or $-b^*$, we can discretize the area around its vertices by an appropriately chosen grid and enumerate all possible near time series; one of them is guaranteed to be sufficiently close to b^* (or $-b^*$), therefore providing an approximate solution whose approximation factor is controlled by the side-length parameter of the grid. To compute the cost of a time series we make use of the following claim, which is implied by Theorem 23 in Section 5.1.

▷ Claim 19. Given a set of time series $P \subset \mathbb{R}^m$ and a time series $b \in \mathbb{R}^l$, one can compute $cost(P, \{b\})$ in time $O(m^{4.5}l^2)$.

This procedure is described in Algorithm 2 and its performance is stated in Theorem 20.

Algorithm 2 $1+\varepsilon$-approximation algorithm for finding a single base curve.

procedure `EpsilonApprox` $(P \subset \mathbb{R}^m,\ l \in \mathbb{N},\ \varepsilon \in \mathbb{R}_{>0})$
  $c \leftarrow$ `ConstantApprox`(P, l) ▷ Algorithm 1
  $\Delta' \leftarrow \mathrm{cost}(P, \{c\})$
  $X \leftarrow \sum_{x\in P} \|x\|_\infty$
  $\delta \leftarrow \frac{\varepsilon}{60}\cdot\frac{\Delta'}{X}$
  $S_i \leftarrow [c_i - \frac{40\Delta'}{X}, c_i + \frac{40\Delta'}{X}]$ for $i \in [l]$
  $\mathcal{C} \leftarrow (\bigcup_{i\in[l]} S_i \cap G_\delta)$, where $G_\delta = \{i\cdot\delta \mid i \in \mathbb{Z}\}$
  Let $b' \in \arg\min_{b\in\mathcal{C}^l} \mathrm{cost}(P, \{b\})$ be arbitrary
  return b'

▶ **Theorem 20.** *Given n time series $P \subset \mathbb{R}^m$ and parameter $l \in \mathbb{N}$, $\varepsilon \in \mathbb{R}_{>0}$, let b^* be an optimal solution for the Fréchet decomposition problem with $k = 1$. Then Algorithm 2 computes in time $O\left(\frac{l}{\varepsilon}\right)^l \cdot O(nm^{4.5}l^2) + \tilde{O}(n^2ml)$ a time series $b' \in \mathbb{R}^l$ such that*

$$\mathrm{cost}(P, \{b'\}) \le (1+\varepsilon)\cdot \mathrm{cost}(P, \{b^*\}).$$

Proof. By Theorem 17 it holds that $\Delta' \in [\mathrm{cost}(P, \{b^*\}), 30\,\mathrm{cost}(P, \{b^*\})]$ and ConstantApprox(P, l) computes c in time $\tilde{O}(n^2ml)$.

By Lemma 18, there exists $w \in \{-1, 1\}$, such that

$$d_F(c, w \cdot b^*) \leq \frac{40\Delta'}{\sum_{x\in P} \|x\|_\infty}.$$

Hence, all vertices of b^* or $-b^*$ lie in intervals $S_i := \left[c_i - \frac{40\Delta'}{\sum_{x\in P}\|x\|_\infty}, c_i + \frac{40\Delta'}{\sum_{x\in P}\|x\|_\infty}\right]$, where c_i, $i \in [l]$ are the vertices of c. Let $\mathcal{C}$ be defined as in the algorithm, which is the area of interest discretized by a grid with side length δ. The algorithm returns the time series b' in $\mathcal{C}^l$ that minimizes the cost. Lemma 18 and the construction of $\mathcal{C}$ imply that there exists a time series $b \in \mathcal{C}^l$ such that $\|b - b^*\|_\infty \leq \delta$. For $x \in P$, let $w_x^* \in \arg\min_{w\in\mathbb{R}} d_F(x, w \cdot b^*)$. The cost of b' is then upper bounded as follows:

$$\begin{aligned}
\mathrm{cost}(P, \{b'\}) &\leq \mathrm{cost}(P, \{b\}) &(1)\\
&= \sum_{x\in P} \min_{w\in\mathbb{R}} d_F(x, w \cdot b)\\
&\leq \sum_{x\in P} d_F(x, w_x^* \cdot b)\\
&\leq \sum_{x\in P} \left(d_F\left(x, w_x^* \cdot b^*\right) + d_F\left(w_x^* \cdot b^*, w_x^* \cdot b\right)\right) &(2)\\
&\leq \mathrm{cost}(P, \{b^*\}) + \sum_{x\in P} w_x^* \delta\\
&\leq \mathrm{cost}(P, \{b^*\}) + 2\delta \sum_{x\in P} \|x\|_\infty &(3)\\
&\leq \mathrm{cost}(P, \{b^*\}) + \frac{\varepsilon}{30}\Delta' &(4)\\
&\leq (1+\varepsilon)\,\mathrm{cost}(P, \{b^*\}),
\end{aligned}$$

where in (2) we use the triangle inequality and in (3) we use Lemma 12.

For each time series in $\mathcal{C}$, we compute the total cost in time $O(nm^{4.5}l^2)$ using Claim 19. The size of $\mathcal{C}$ is in $O\left(\frac{l}{\varepsilon}\right)^l$; hence, the overall running time is $O\left(\frac{l}{\varepsilon}\right)^l \cdot O(nm^{4.5}l^2) + \tilde{O}(n^2ml)$. ◀

By first applying Theorem 8 on the input time series we obtain the following Corollary.

▶ **Corollary 21.** *Given n time series $P \subset \mathbb{R}^m$ and constants $l \in \mathbb{N}$ and $\varepsilon \in (0, 1)$. Let b^* be an optimal solution for the Fréchet decomposition problem with $k = 1$. One can compute in time $\tilde{O}(n^2 + nm)$ a time series $b' \in \mathbb{R}^l$ such that*

$$\mathrm{cost}(P, \{b'\}) \leq (1+\varepsilon) \cdot \mathrm{cost}(P, \{b^*\}).$$

5 Exact algorithms

5.1 Projection Distance Problem

In this section, we give an algorithm for the projection distance problem. Missing proofs of this section can be found in the corresponding section of the full version of the paper [20]. We start by solving the problem for the case that the weight vector is fixed. Note that in this case we can assume without loss of generality that the weight vector is the vector of all 1s since we can scale each base curve by its optimal weight and use the resulting set of base curves instead. We solve the problem using dynamic programming. Let $x \in \mathbb{R}^m$ and $b_1, \ldots, b_k \in \mathbb{R}^l$. For a vector $v = (v_1, \ldots, v_r)$ we will use $v^{(i)} = (v_1, \ldots, v_i)$ to be the

i-dimensional vector that agrees with v on the first i coordinates. Additionally for b_j with $1 \leq j \leq k$ we use the notation $(b_j)_i$ to address its i-th coordinate. For the dynamic program, we use $D \subset \mathbb{R}^{m \times l^k}$ to denote a multidimensional array of which each entry $D[i_0, i_1, \ldots, i_k]$ will store the target value

$$\min_{M_0,\ldots,M_k} \|M_0 x^{(i_0)} - \sum_{j=1}^{k} M_j b_j^{(i_j)}\|_\infty$$

Intuitively, the target value is the Fréchet distance of the prefix of x to the (unit weight) span of the prefixes of the base vectors (cf. Observation 5). The subproblems solved by the dynamic program are parametrized by the lengths of these prefixes. To define the dynamic program computing these entries, we use the following recursion:

$$D[i_0, \ldots, i_k] = \max\Big\{\min_{(h_0,\ldots,h_k) \in H_{i_0,\ldots,i_k}} D[i_0 - h_0, \ldots, i_k - h_k], |x_{i_0} - \sum_{j=1}^{k} (b_j)_{i_j}|\Big\},$$

where $H_{i_0,\ldots,i_k} = \big\{(h_0, \ldots, h_k) \in \{0,1\}^{k+1} \ : \ i_j - h_j \geq 1 \text{ for } 0 \leq j \leq k\big\} \setminus \big\{(0, \ldots, 0)\big\}$ is the set of possible movements among the involved time series and we set $D[1, \ldots, 1] = |x_1 - \sum_{j=1}^{k} (b_j)_1|$. We may abbreviate notation and write $D[\vec{i}]$ and $H_{\vec{i}}$ for $\vec{i} = (i_0, \ldots, i_k)$. The dynamic program is given in Algorithm 3. The proof builds upon the reasoning behind the standard DP formulation of the discrete Fréchet distance computation in [23].

Algorithm 3 Solving the projection distance problem when the weight vector is fixed.

```
1: procedure Projection (x ∈ ℝ^m, {b_1, ..., b_k} ⊂ ℝ^l)
2:     Initialize D ⊂ ℝ^{m×l^k}
3:     for i⃗ ∈ [m] × [l]^k in lexicographical order do
4:         if i⃗ = (1, ..., 1) then
5:             D[i⃗] ← |x_1 − Σ_{j=1}^k (b_j)_1|
6:         else
7:             D[i⃗] ← max { min_{h∈H_i⃗} D[i⃗ − h], |x_{i_0} − Σ_{j=1}^k (b_j)_{i_j}| }
8:     return D
```

Given the computed array D of Algorithm 3 one can reconstruct optimal traversal matrices with a standard dynamic programming technique by starting from the top-right corner of D and tracing back to the origin, which decisions have led to the solution. Here we only compute the optimal traversal matrices implicitly by constructing their traversal identifiers to save space and time, since the explicit matrix construction is not necessary for our purposes. The running time of this procedure is dominated by the computation of the table D yielding the following theorem.

▶ **Theorem 22.** *Given time series $x \in \mathbb{R}^m$, $b_1, \ldots, b_k \in \mathbb{R}^l$, one can compute in $O((2l)^k m)$ time, traversal identifiers of traversal matrices $M_0, \ldots, M_k$ that minimize $\|M_0 x - \sum_{j=1}^{k} M_j b_j\|_\infty$.*

We now solve the projection distance problem when the weights are not fixed.

▶ **Theorem 23.** *Given a time series $x \in \mathbb{R}^m$ and k time series $B = \{b_1, \ldots, b_k\} \subset \mathbb{R}^l$, one can compute $d_F(x, \mathrm{span}_{\mathrm{F}}(B))$ in $O(2^k m^{(2k+2.5)} l^{2k^3})$ time.*

Proof. Our algorithm initially constructs an arrangement of hyperplanes, such that each cell of the arrangement defines a subset of $\mathbb{R}^k$ corresponding to weight values that all lead to the same sequence of decisions taken by Algorithm 3. Having such an arrangement then allows us to treat our problem in two steps: we compute the best traversal matrices within each cell, and for each set of such traversal matrices we compute the best possible weights. Let $M_0^*, \ldots, M_k^*$ be optimal traversal matrices and $w_1^*, \ldots, w_k^* \in \mathbb{R}$ be optimal weights:

$$d_F(x, \mathrm{span_F}(B)) = \|M_0^* x - \sum_{j=1}^{k} w_j^* M_j^* b_j\|_\infty.$$

We start by defining hyperplanes that encode relations of distances between any vertex of the input time series and any candidate (matched) vertex of its projection. Given $\vec{i} = (i_0, \ldots, i_k) \in [m] \times [l]^k, \vec{j} = (j_0, \ldots, j_k) \in [m] \times [l]^k$, for each $\sigma \in \{-1, +1\}$ we define

$$s_\sigma(\vec{i}, \vec{j}) := \left\{ (w_1, \ldots, w_k) \in \mathbb{R}^k : x_{i_0} - \sum_{y=1}^{k} w_y (b_y)_{i_y} = \sigma \cdot (x_{j_0} - \sum_{y=1}^{k} w_y (b_y)_{j_y}) \right\}$$

and $s(\vec{i}, \vec{j}) := \left\{ s_\sigma(\vec{i}, \vec{j}) \mid \sigma \in \{-1, +1\} \right\}$. Essentially, the arrangement of hyperplanes $\mathcal{A}(s(\vec{i}, \vec{j}))$ partitions $\mathbb{R}^k$ into cells, such that any weight vector within the same cell induces the same sign on $x_{i_0} - \sum_{y=1}^{k} w_y (b_y)_{i_y} - \sigma \cdot (x_{j_0} - \sum_{y=1}^{k} w_y (b_y)_{j_y})$, for any $\sigma \in \{-1, +1\}$, therefore inducing the same sign on $|x_{i_0} - \sum_{y=1}^{k} w_y (b_y)_{i_y}| - |x_{j_0} - \sum_{y=1}^{k} w_y (b_y)_{j_y}|$.

Next, we define the set of hyperplanes

$$\mathcal{S} := \bigcup_{\vec{i}, \vec{j} \in [m] \times [l]^k} s(\vec{i}, \vec{j}).$$

For each cell F of the arrangement $\mathcal{A}(\mathcal{S})$ there exists a common ordering of the values $|x_{i_0} - \sum_{y=1}^{k} w_y (b_y)_{i_y}|$ over all $i \in [m] \times [l]^k$. This implies that each cell F of the arrangement $\mathcal{A}(\mathcal{S})$ is a subset of $\mathbb{R}^k$, for which, for any $(w_0, \ldots, w_k) \in \mathrm{cl}(F)$, the evaluations made by Algorithm 3 in Line 7, throughout all iterations, are fixed. Hence, by the correctness of Algorithm 3, there are traversal matrices $M_0, \ldots, M_k$ such that for any $(w_0, \ldots, w_k) \in \mathrm{cl}(F)$, $M_0, \ldots, M_k$ minimize $\|M_0 x - \sum_{i=1}^{k} w_i M_i b_i\|_\infty$. Additionally, by the definition of $\mathcal{S}$ there is a cell F^* of $\mathcal{A}(\mathcal{S})$ such that $M_0^*, \ldots, M_k^*$ minimizes $\|M_0 x - \sum_{i=1}^{k} w_i M_i b_i\|_\infty$ for any $(w_0, \ldots, w_k) \in \mathrm{cl}(F^*)$. We then compute the minimizing weights of $\mathrm{cl}(F)$ as follows. Let $c_0, \ldots, c_k$ be the traversal identifiers of Theorem 22, let t be their common dimension, and let $M_0, \ldots, M_k$ be the corresponding optimal traversal matrices. Finally let $I_F := \{((c_0)_r, \ldots, (c_k)_r) \mid \text{ for } r \in [t]\}$. We have

$$\forall w \in \mathrm{cl}(F): \ \|M_0 x - \sum_{i=1}^{k} w_i M_i b_i\|_\infty = \max_{(i_0, \ldots, i_k) \in I_F} |x_{i_0} - \sum_{j=1}^{k} w_j (b_j)_{i_j}|.$$

Moreover, by the definition of $\mathcal{S}$, for each cell F of $\mathcal{A}(\mathcal{S})$, there is a uniquely defined function $\sigma_F: \ [m] \times [l]^k \mapsto \{-1, 1\}$ such that

$$\forall w \in \mathrm{cl}(F): \ \|M_0 x - \sum_{i=1}^{k} M_i w_j b_i\|_\infty := \max_{(i_0, \ldots, i_k) \in I_F} \sigma_F(i_0, \ldots, i_k) \cdot (x_{i_0} - \sum_{j=1}^{k} w_j (b_j)_{i_j}).$$

Formally, we need to solve the following optimization problem:

$$\min_{w \in \mathrm{cl}(F)} \max_{(i_0, \ldots, i_k) \in I_F} \sigma_F(i_0, \ldots, i_k) \cdot (x_{i_0} - \sum_{j=1}^{k} w_j (b_j)_{i_j}).$$

This can be written as a linear program as follows:

$$\text{objective: } \min_{r \in \mathbb{R}} r$$

$$\text{constraints: } \forall (i_0, \ldots, i_k) \in I_F : \ \sigma_F(i_0, \ldots, i_k) \cdot (x_{i_0} - \sum_{j=1}^{k} w_j (b_j)_{i_j}) \le r$$

$$(w_1, \ldots, w_k) \in \mathrm{cl}(F).$$

Doing so for each cell F of $\mathcal{A}(\mathcal{S})$, we return the best weights w and $M_0, \ldots, M_k$ encountered this way, implying the correctness. Note that the linear program is bounded, since by the definition of σ_F, the left-hand side terms of the constraint inequalities are non-negative; hence r is non-negative.

Next, we analyze the running time of this procedure. The arrangement $\mathcal{A}(\mathcal{S})$ contains $O(m^2 l^{2k})$ hyperplanes. Therefore, the number of cells is in $O(m^{2k} l^{2k^2})$ and $\mathcal{S}$ can be constructed using the Algorithm of Edelsbrunner, O'Rourke, and Seidel [22] in time $O(m^{2k} l^{2k^2})$. For each cell we apply Corollary 22, which requires $O(2^k m l^k)$ time. The running time needed to find optimal weights of a cell is in $O((km)^{2.5})$ by [40] since the LP contains $O(km)$ variables. In total, this is $O(m^{2k} l^{2k^2} (2^k m l^k + (km)^{2.5})) \subset O(2^k m^{(2k+2.5)} l^{2k^3})$. ◀

5.2 Fréchet Decomposition Problem

The results of Section 5.1 imply a solution for the Fréchet decomposition problem when the base curves are chosen from a predefined finite set of candidates C. By simply enumerating all k-combinations from C, and computing their costs using Theorem 23, we obtain the following result.

▶ **Corollary 24.** *Let k and $l \in \mathbb{N}$ be constants. Given a set of n time series $P \subset \mathbb{R}^m$, $k \in \mathbb{N}$, a set of time series $C \subset \mathbb{R}^l$ with $|C| \ge k$, we can compute a set of k time series $B \subseteq C$ that minimizes cost(P, B) in time $O\left(|C|^k n m^{2k+2.5}\right)$.*

By first applying Theorem 8 on the input, we obtain the following corollary.

▶ **Corollary 25.** *Let $k, l \in \mathbb{N}$ and $\varepsilon \in (0, 1)$ be constants. Given a set of n time series $P \subset \mathbb{R}^m$, a set of time series $C \subset \mathbb{R}^l$ with $|C| \ge k$ and $B \subseteq C$ with $|B| = k$. We can compute a set of k time series that approximates cost(P, B) within a factor of $(1 + \varepsilon)$ in time $\tilde{O}\left(nm + |C|^k n\right)$, where the $\tilde{O}$-notation hides terms that are polylogarithmic.*

References

1 Amir Abboud and Karl Bringmann. Tighter Connections Between Formula-SAT and Shaving Logs. In *45th International Colloquium on Automata, Languages, and Programming*, volume 107, pages 8:1–8:18, 2018. doi:10.4230/LIPIcs.ICALP.2018.8.

2 Pankaj K. Agarwal, Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. Computing the Discrete Fréchet Distance in Subquadratic Time. In *SIAM Journal on Computing*, volume 43, pages 429–449, 2014. doi:10.1137/130920526.

3 Helmut Alt and Michael Godau. Computing the Fréchet distance between two polygonal curves. In *International Journal of Computational Geometry & Applications*, volume 5, pages 75–91, 1995. doi:10.1142/S0218195995000064.

4 Rinat Ben Avraham, Haim Kaplan, and Micha Sharir. A faster algorithm for the discrete Fréchet distance under translation. In *Computing Research Repository*, 2015. arXiv:1501.03724.

5 Sergey Bereg, Minghui Jiang, Wencheng Wang, Boting Yang, and Binhai Zhu. Simplifying 3D Polygonal Chains Under the Discrete Fréchet Distance. In *LATIN 2008: Theoretical Informatics, 8th Latin American Symposium*, volume 4957, pages 630–641, 2008. doi:10.1007/978-3-540-78773-0_54.

6 Karl Bringmann. Why Walking the Dog Takes Time: Frechet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails. In *55th IEEE Annual Symposium on Foundations of Computer Science*, pages 661–670, 2014. doi:10.1109/FOCS.2014.76.

7 Karl Bringmann, Marvin Künnemann, and André Nusser. Discrete Fréchet Distance under Translation: Conditional Hardness and an Improved Algorithm. In *ACM Transactions on Algorithms*, pages 25:1–25:42, 2021. doi:10.1145/3460656.

8 Karl Bringmann and Wolfgang Mulzer. Approximability of the discrete Fréchet distance. In *Journal of Computational Geometry*, volume 7, pages 46–76, 2016. doi:10.20382/JOCG.V7I2A4.

9 Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2922–2938, 2019. doi:10.1137/1.9781611975482.181.

10 Kevin Buchin, Anne Driemel, and Martijn Struijs. On the Hardness of Computing an Average Curve. In *17th Scandinavian Symposium and Workshops on Algorithm Theory*, volume 162, pages 19:1–19:19, 2020. doi:10.4230/LIPIcs.SWAT.2020.19.

11 Kevin Buchin, Tim Ophelders, and Bettina Speckmann. SETH Says: Weak Fréchet Distance is Faster, but only if it is Continuous and in One Dimension. In *30th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2887–2901, 2019. doi:10.1137/1.9781611975482.179.

12 Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k,ℓ)-Median Clustering for Polygonal Curves. In *ACM Transactions on Algorithms*, volume 19, pages 4:1–4:32, 2023. doi:10.1145/3559764.

13 Maike Buchin and Dennis Rohde. Coresets for (k,ℓ)-Median Clustering Under the Fréchet Distance. In *Algorithms and Discrete Applied Mathematics*, pages 167–180, 2022. doi:10.1007/978-3-030-95018-7_14.

14 Siu-Wing Cheng and Haoqiang Huang. Curve Simplification and Clustering under Fréchet Distance. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 1414–1432, 2023. doi:10.1137/1.9781611977554.CH51.

15 Vincent Cohen-Addad, Andrew Draganov, Matteo Russo, David Saulpic, and Chris Schwiegelshohn. A Tight VC-Dimension Analysis of Clustering Coresets with Applications. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 4783–4808, 2025. doi:10.1137/1.9781611978322.162.

16 Marco Cuturi and Mathieu Blondel. Soft-DTW: a Differentiable Loss Function for Time-Series. In *34th International Conference on Machine Learning*, volume 70, pages 894–903, 2017. URL: http://proceedings.mlr.press/v70/cuturi17a.html.

17 Marco Cuturi, Jean-Philippe Vert, Øystein Birkenes, and Tomoko Matsui. A Kernel for Time Series Based on Global Alignments. In *International Conference on Acoustics, Speech, and Signal Processing*, pages 413–416, 2007. doi:10.1109/ICASSP.2007.366260.

18 Anne Driemel, Jan Höckendorff, Ioannis Psarros, and Christian Sohler. Fréchet PCA. Software, DFG - 459420781 (https://gepris.dfg.de/project/459420781), swhId: swh:1:dir:42897eeb29e82508265e89725746120e29ec7177 (visited on 2026-08-13). URL: https://github.com/jahoec/PCA-on-Curves, doi:10.4230/artifacts.27658.

19 Anne Driemel, Jan Höckendorff, Ioannis Psarros, and Christian Sohler. A near-linear time approximation scheme for (k, ℓ)-median clustering under discrete Fréchet distance. In *Computing Research Repository*, 2025. doi:10.48550/arXiv.2508.07008.

20 Anne Driemel, Jan Höckendorff, Ioannis Psarros, and Christian Sohler. Time series decomposition using the fréchet distance. In *Computing Research Repository*, 2026. arXiv:2607.04397.

21 Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *27th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–785, 2016. doi:10.1137/1.9781611974331.ch55.

22 Herbert Edelsbrunner, Joseph O'Rourke, and Raimund Seidel. Constructing Arrangements of Lines and Hyperplanes with Applications. In *SIAM Journal on Computing*, volume 15, pages 341–363, 1986. doi:10.1137/0215024.

23 Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. *Christian Doppler Labor für Expertensyteme, Technical University of Vienna*, 1994. URL: https://www.kr.tuwien.ac.at/staff/eiter/et-archive/files/cdtr9464.pdf.

24 Abdel Hannachi, Ian T Jolliffe, and David B Stephenson. Empirical orthogonal functions and related techniques in atmospheric science: A review. In *International Journal of Climatology*, volume 27, pages 1119–1152, 2007. doi:10.1002/JOC.1499.

25 Paul Honeine and Cédric Richard. Preimage Problem in Kernel-Based Machine Learning. In *IEEE Signal Processing Magazine*, volume 28, pages 77–88, 2011. doi:10.1109/MSP.2010.939747.

26 John D Horel. Complex principal component analysis: Theory and examples. In *Journal of Applied Meteorology and Climatology*, volume 23, pages 1660–1673, 1984. doi:10.1175/1520-0450(1984)023<1660:CPCATA>2.0.CO;2.

27 Minghui Jiang, Ying Xu, and Binhai Zhu. Protein Structure-structure Alignment with Discrete FrÉchet Distance. In *Journal of Bioinformatics and Computational Biology*, volume 6, pages 51–64, 2008. doi:10.1142/S0219720008003278.

28 Christian Jutten and Jeanny Hérault. Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture. In *Signal Processing*, volume 24, pages 1–10, 1991. doi:10.1016/0165-1684(91)90079-X.

29 Alois Kneip and James O Ramsay. Combining registration and fitting for functional models. In *Journal of the American Statistical Association*, volume 103, pages 1155–1165, 2008. doi:10.1198/016214508000000517.

30 Sungwon Lee and Sungkyu Jung. Combined analysis of amplitude and phase variations in functional data. In *Computing Research Repository*, 2016. arXiv:1603.01775.

31 Edward N Lorenz. *Empirical orthogonal functions and statistical weather prediction*. Massachusetts Institute of Technology, Department of Meteorology Cambridge, 1956.

32 Sebastian Mika, Bernhard Schölkopf, Alexander J. Smola, Klaus-Robert Müller, Matthias Scholz, and Gunnar Rätsch. Kernel PCA and De-Noising in Feature Spaces. In *Advances in Neural Information Processing Systems 11*, pages 536–542, 1998. URL: http://papers.nips.cc/paper/1491-kernel-pca-and-de-noising-in-feature-spaces.

33 Axel Mosig and Michael Clausen. Approximately matching polygonal curves with respect to the Fre'chet distance. In *Computuational Geometry*, volume 30, pages 113–127, 2005. doi:10.1016/J.COMGEO.2004.05.004.

34 Abhinandan Nath and Erin Taylor. k-Median clustering under discrete Fréchet and Hausdorff distances. In *Journal of Computational Geometry*, volume 12, pages 156–182, 2021. doi:10.20382/JOCG.V12I2A8.

35 Maurice Bertram Priestley. *Spectral analysis and time series*, volume 890. Academic press London, 1981.

36 James O Ramsay and Xiaochun Li. Curve Registration. In *Journal of the Royal Statistical Society Series B: Statistical Methodology*, volume 60, pages 351–363, 1998. doi:10.1111/1467-9868.00129.

37 Robert H Shumway and David S Stoffer. *Time series analysis and its applications*, volume 3. Springer, 2000.

38 Koh Takeuchi, Masaaki Imaizumi, Shunsuke Kanda, Yasuo Tabei, Keisuke Fujii, Ken Yoda, Masakazu Ishihata, and Takuya Maekawa. Fréchet Kernel for Trajectory Data Analysis. In *29th International Conference on Advances in Geographic Information Systems*, pages 221–224, 2021. doi:10.1145/3474717.3483949.

39 Bernd Uebbing, Jan Höckendorff, Caroline Jungheim, Anne Driemel, Christian Sohler, and Jürgen Kusche. An alternative to PCA utilizing Dynamic Time Warping. Technical report, Copernicus Meetings, 2024.

40 Pravin M. Vaidya. Speeding-Up Linear Programming Using Fast Matrix Multiplication (Extended Abstract). In *30th Annual Symposium on Foundations of Computer Science*, pages 332–337, 1989. `doi:10.1109/SFCS.1989.63499`.

41 Jane-Ling Wang, Jeng-Min Chiou, and Hans-Georg Müller. Functional Data Analysis. In *Annual Review of Statistics and its application*, volume 3, pages 257–295, 2016. `doi:10.1146/annurev-statistics-041715-033624`.

42 Kongming Wang and Theo Gasser. Alignment of curves by dynamic time warping. In *The annals of Statistics*, volume 25, pages 1251–1276, 1997. `doi:10.1214/aos/1069362747`.

43 Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. In *Chemometrics and intelligent laboratory systems*, volume 2, pages 37–52, 1987. `doi:10.1016/0169-7439(87)80084-9`.