On clustering and related problems on curves under the Fréchet distance

Dissertation

zur Erlangung des Grades eines

D o k t o r s d e r N a t u r w i s s e n s c h a f t e n

der Technischen Universität Dortmund
an der Fakultät für Informatik

von
Amer Krivošija

Dortmund
2021

Tag der mündlichen Prüfung:	11. Februar 2021
Dekan:	Prof. Dr. Gernot A. Fink
Gutachter:	Prof. Dr. Christian Sohler
	Prof. Dr. Anne Driemel
	Prof. Dr. Erich Schubert

Abstract

Sensor measurements can be represented as points in $\mathbb{R}^d$. Ordered by the time-stamps of these measurements, they yield a time series, that can be interpreted as a polygonal curve in the d-dimensional ambient space. In this thesis we study several fundamental computational tasks on curves: clustering, simplification, and embedding.

The Fréchet distance is a popular distance measure for curves, in its continuous and discrete version. It is a distance measure of choice should the inner structure of the curves be observed. The Fréchet distance lends itself naturally to the computational tasks we investigate, in the corresponding metric spaces. One of the limitations is the inherent complexity of the computation of the Fréchet distance, as it is widely believed that no algorithms exist to compute either the discrete or the continuous Fréchet distance between two curves with m vertices each, in the subquadratic running time in m. The number of the vertices is called *complexity* of the curve.

In this thesis we focus on curves in the one-dimensional ambient space $\mathbb{R}$. We study the problem of clustering of the curves in one-dimensional ambient space under the Fréchet distance, in particular, the following variations of the well-known k-center and k-median problems. Given is a set P of n curves in one-dimensional ambient space, each of complexity at most m. Our goal is to find k one-dimensional curves, not necessarily from P, that we call *cluster centers* and that each has complexity at most ℓ. In the (k, ℓ)-center problem, the maximum distance of an element of P to its nearest cluster center is minimized. In the (k, ℓ)-median problem, the sum of these distances is minimized. We show that both problems are **NP**-hard under both the discrete and the continuous Fréchet distance, if k is part of the input.

Under the continuous Fréchet distance, we give $(1 + \varepsilon)$-approximation algorithms for both (k, ℓ)-center and (k, ℓ)-median problem, with running time near-linear in the input size for constant ε, k and ℓ. Our techniques yield constant-factor approximation algorithms for the observed problems under the discrete Fréchet distance.

To obtain the $(1 + \varepsilon)$-approximation algorithms for the clustering problems under the continuous Fréchet distance, we develop a new simplification technique on one-dimensional curve. Our simplifications, called *δ-signatures*, provide the "shape" of the curve. The parameter δ relates to the minimum length of the edges of the simplified curve. The signatures always exist, and we can compute them efficiently.

We also study the problem of embedding of the (discrete and continuous) Fréchet distance into one-dimensional ambient space. More precisely, we study distortion of the probabilistic embedding that results from projecting the curves onto a randomly chosen line. We show that, in the worst case and under reasonable assumptions, the discrete Fréchet distance

between two polygonal curves of complexity m in $\mathbb{R}^d$, where $d \in \{2, 3, 4, 5, 6, 7\}$, degrades by a factor linear in m with constant probability. We show upper and lower bounds on the distortion.

Sensor measurements can also define a discrete distribution over possible locations of a point in $\mathbb{R}^d$. Then, the input consists of n probabilistic points. We study the probabilistic 1-center problem in Euclidean space $\mathbb{R}^d$, also known as the probabilistic smallest enclosing ball (pSEB) problem. Our main objective is to improve the best existing algorithm for the pSEB problem by reducing its exponential dependence on the dimension to linear.

To do so, we study the deterministic *set median* problem, which is a variant of the median problem for a collection of point sets in high dimensions. The set median problem generalizes the 1-median as well as the (probabilistic) 1-center problems. We present a $(1 + \varepsilon)$-approximation algorithm for the set median problem, using a novel combination of sampling techniques for clustering problems in metric spaces with the framework of stochastic subgradient descent.

Our $(1+\varepsilon)$-approximation algorithm for the pSEB problem takes $O\left(\left(dn/\varepsilon^4\right) \cdot \log^2\left(1/\varepsilon\right)\right)$ time. As a result, the pSEB algorithm becomes applicable to shape fitting problems in Hilbert spaces of unbounded dimension using kernel functions. We present an exemplary application by extending the support vector data description (SVDD) shape fitting method to the probabilistic case. This is done by simulating the pSEB algorithm implicitly in the feature space induced by the kernel function.

Contents

1 Introduction

Sensors and other measuring devices are part of everyone's day-to-day life. We actively contribute to many measuring processes, with or without being aware of it, for example, to our electronic fingerprint by browsing online-shops or wearing a tracking device, like a cell phone or a smart watch. Some measurements result from observing phenomena that happen independently of us, for example those observed at IceCube Neutrino Observatory at Amundsen–Scott South Pole Station, or the air temperature and the wind speed and direction. There are phenomena we can partially influence, such as the stock market and commodities prices, or annual migration routes of various bird species.

These measurements can be one-dimensional, e.g. represented by an integer or by a real number, or multi-dimensional, where the number of dimensions can be extremely high. Each of the measurements can be combined with a time stamp. From recording the measurements we can learn about the behavior of the observed object or phenomenon, and recognize their habits, customs, or patterns. Based on the gained knowledge, we hope to be able to predict the future behavior. If we model the measurements as points and/or the trajectories that describe changes of the measured values as polygonal lines, then the problems we are solving are *computational geometry problems*.

The sensors generate a vast amount of data every day. It is not always possible to store everything we measure, or even to know the amount of information in advance. When the data is stored, often only a limited number of passes over the data is allowed. It is of a particular interest for scientists, business people, governments, etc. to be able to *efficiently* analyze the recorded data. That can be done by identifying an "important" part of the data, or by simplifying the input, or it is often enough to pick a small random sample from the collected data, and make a good prediction for the whole set based on the chosen sample.

1.1 Choice of the objects observed

On deterministic points and sets

The measured position or the state of some object (e.g. a person that wears a smart watch, or a bird that has a tracking ring) can be described by a point in the space $\mathbb{R}^d$, where d is a positive integer. The dissimilarity (or the relative position) of two points can be measured in various manners, most often by the Euclidean distance, which "naturally" describes the straight line distance between them. One may ask, why is this distance measure *"more natural"* than some other, e.g. well-known Manhattan distance? Indeed, the Manhattan distance follows the geometry of city streets that form a rectangular grid, thus a person with a smart watch that commutes from point A to point B has to walk along the streets in order to reach her destination. A bird, on the other hand, can still use the Euclidean straight path on its way from A to B. In this thesis, our choice of the distance measure between two points in $\mathbb{R}^d$ is the Euclidean distance.

A similar question on "naturality" can be posed for the distance between a point and a set of points in $\mathbb{R}^d$. A flying bird that is currently over the sea and wants to reach the shore as fast as possible will choose the *closest* point on the shore. This behavior represents the Hausdorff distance (cf. the definition in Subsection 2.5.2). Sometimes one does not want to know the nearest, but the *farthest* neighbor. This can be illustrated by an example given by Pagh *et al.* [149]: if an online-shop-customer purchased a product, it is reasonable to offer her additional products, that are related to the purchased, but rather far from it, to increase the probability of further (successful) shopping. We address such measure within the set median problem in Section 6.2.

On curve representation and dissimilarity measures

Next, we assume that we are no longer observing just a current position of a point, but also the trajectories that are made by tracing the point, e.g. GPS tracking of a person during one day. If a sequence of points $w_1, \ldots, w_m$ in $\mathbb{R}^d$ is connected by straight line segments in the order of the indices of the points, we obtain a polygonal curve in the d-dimensional ambient space. A polygonal curve is a standard representation for the continuous real-life curves in the computational geometry.

In the other computer science (sub)communities, e.g. in machine learning, one works with *time series* instead of curves. Time series are sequences of discrete measurements of a continuous signal, e.g. $S = (w_1, t_1), \ldots, (w_m, t_m)$, consisting of paired values of measurements w_j and time stamps t_j, $1 \leq j \leq m$. Examples of data that are often represented as time series include stock market values, electrocardiograms, temperature,

the number of newly infected persons per day, and the hourly requests of a webpage. Note that all of these measurements are one-dimensional, but they can be multi-dimensional as well, e.g. voters' polling results, vital parameters of a patient, etc.

The time stamp of a point can be observed as an additional dimension of the point. Then, a time series would have one dimension more than a curve, based on the same measurements. However, in this thesis, we decide that a dissimilarity measure should consider only the ordering of the measurements/points, and ignore the explicit time stamps. Then the two notions are equivalent *and of same dimension*. It is the shape of curves that counts, and not the "speed of points" along the curves.

A common approach to measure the dissimilarity between the two curves is to treat the vertices of the curve (time series data) as coordinates of a point in high-dimensional space, i.e. the curve $w_1, \ldots, w_m$ and the time series $(w_1, t_1), \ldots, (w_m, t_m)$ are treated as a point $(w_1, \ldots, w_m)$ in the m-dimensional Euclidean space. Using this simple interpretation of the data, to process the input in a desired way, any algorithm for points in high-dimensional space can be applied. Although this is a common practice (e.g. in the work of Liao [131]), it has many limitations. One major drawback is the requirement that all the curves must have the same number of vertices, and the measurements must be regular and synchronized. In particular, for multiple sensors, the latter requirement is often hard to achieve.

Another option to measure the dissimilarity between the two curves is to treat the curves as *sets*, and thus, to use the Hausdorff distance. Such approach seems simple, but it completely ignores the inner structure of the curves. Namely, this approach could identify the two curves as similar (e.g. two taxi customers are taking the same road), based only on the fact that they consist of similar vertices, but without observing the time stamps order (e.g. that the routes are in opposite directions). See Subsection 2.5.2 for an example.

We assume that it is "natural" to observe the inner structure of the curves when measuring their dissimilarity, instead of simply observing the endpoints of the curves, the closest or the farthest pair of points on the curves, etc. These goals are achieved if we choose the (discrete or continuous) *Fréchet distance*. The continuous Fréchet distance was proposed by Fréchet [85] in 1906, in the context of the study of general metric spaces. The discrete Fréchet distance was introduced by Eiter and Mannila [75] in 1994.

The Fréchet distance intuitively describes the minimal cost of transforming one curve into another, where the cost measure of the transformation is the *maximum distance* between two points mapped to each other by the transformation of the curves[1] (for formal definitions see Subsection 2.5.2). For the continuous Fréchet distance, the complete curves

[1] In the literature, the intuitive introduction to the Fréchet distance utilizes a man-and-dog-on-the-leash metapher. We opt to avoid that.

have to be mapped to each other, while respecting the order of the points on the curves. In handwriting identification [164], the letters can be of different size or font, but the similarity between respective parts of a letter is decisive.

In one-dimensional ambient space, the interpretation of the continuous Fréchet distance has another aspect. For two monotone curves in $\mathbb{R}$, the continuous Fréchet distance equals the maximum of the distances between the start- and endpoints, pairwise. This also implies, that the continuous Fréchet distance between two functions is completely determined by the sequence of local extrema ordered by their indices/time stamps. Thus, when considering the continuous Fréchet distance of the curves, we view a one-dimensional curve as a specification of the sequence of local extrema of the curve.

However, the continuous transformation measure between the two curves does not always make sense. In biology, the proteins consist of amino acids, each containing α-carbon atoms. The carbon atoms can be observed as vertices, whose order determines the structure of a protein [176]. If we would measure the dissimilarity of two proteins by the continuous Fréchet distance, two arbitrary points on the curves (proteins) could be mapped to each other, which is biologically meaningless. Here, it is natural to consider only vertices of the two curves for the transformation of one curve into another, in which case we have the discrete Fréchet distance.

The cost of transforming one curve into another is sometimes not evaluated over maximum, but over the *sum of the distances* between the mapped vertices. This dissimilarity measure related to the discrete Fréchet distance is called the *dynamic time warping* (DTW). DTW was (presumably) first time introduced in the context of speech discrimination and aligning distorted speech signals by Vintsyuk [173] in 1968, and has since been popularized in the field of data mining (cf. [62, 142]). The process of traversing the curves with varying speeds, while summing the dissimilarity measure between the mapped vertices, is referred to as "time-warping" in this context. DTW is known for its universality, and is applied to describe various phenomena, such as chromosomes [129], electrocardiogram frames [104], fingerprints [122], signature comparison [151], etc. However, DTW has some disadvantages against the Fréchet distance. For this work, the most important one is that DTW is not a metric, while both continuous and discrete Fréchet distance are (pseudo)-metrics.

Both versions of the Fréchet distance, as well as the dynamic time warping distance can be computed in worst-case (roughly) quadratic time in the number of vertices of the curves by applying dynamic programming (cf. [5, 40, 88]). On the other side, algorithms to compute any of these three distance measures in strongly subquadratic time, even for one-dimensional curves, do not exist under widely believed complexity theoretic assumptions (cf. [31, 33, 37]). The faster computation is possible, if additional assumptions on the input curves are made.

For the detailed listing of the results see Subsection 2.5.3. The fast computing of these distance measures in practice is also an active research topic (cf. [35, 36, 119]), and is of independent interest, which is out of scope of this thesis.

On probabilistic point distributions

The measurements of positions or quantities of a point in $\mathbb{R}^d$ can reveal that the locations (values, amounts) get repeated, and that the frequencies of their appearing, or not appearing, at certain locations are known. Then one could be interested in discovering the patterns that are valid in expectation. For example, a taxi customer requires more often to be picked up from her workplace, than from her sport club, for a ride home. But it is also possible that she does not call on some days at all. A taxi company would like to know where ideally to place a car, so that the expected length of a route to the customer is as short as possible.

The problems can also be high-dimensional, e.g. in combustion, a flame reactor deals with hundreds of reactions and species, resulting from multiple chemical components. The reactions occur and the species appear under certain conditions and in various quantities. They all have various influence onto the combustion scenario. It is of crucial importance for a simulation, to identify the representative reactions/species, such that the expected result is a good approximation of a real combustion in the reactor, that would be too expensive to compute with the detailed chemical data [161, 162].

Therefore, if we assign to a point a discrete probability distribution over the locations in $\mathbb{R}^d$ where the point can appear, or be "not present", at a moment, we have *probabilistic points*. A realization of a set of probabilistic points is achieved when for each point one location is picked. Then, a probabilistic optimization problem is to minimize the objective function *in expectation* over all possible realizations of the probabilistic points. Note that the parameters of the observed problems are: the number of distributions (say n), the maximum number of locations in a distribution (say z), the number of realizations (say N), and the dimensionality of the ambient space d. For formal definitions see Subsection 6.1.1.

1.2 Choice of the problems observed

1.2.1 Input reduction

To model a problem on curves, we begin with two parameters: the number of the curves (say n), and the maximum number of vertices of each curve, called *complexity*, that we denote with m. We consider two input reduction techniques: curve simplification and

metric embedding. However, it is convenient to make a realistic assumption on the input, that simplifies the modelling and the computation.

The realistic analysis of problems on curves is a popular research topic [64]. There are many competing models for a problem simplification through the realistic assumptions, where each of the models has an argument of "being closer to reality". For one of the problems (in Chapter 5), we utilize the concept of *c-packedness*, introduced by Driemel, Har-Peled and Wenk [66], as it appears that this model attracts the most attention in the computational geometry community (cf. [6, 34, 65, 98]). We say for a curve to be c-packed, if its length within each ball in the ambient space is at most c times the radius of the ball (cf. Definition 2.30).

Two alternative models were introduced by Alt, Knauer and Wenk [15]: the *c-straightness*, where the length of a curve between two points is at most c times the Euclidean distance between these points; and the *c-boundedness*, where for each two points x and y on the curve, the part of the curve between x and y is covered by the balls of radius $c/2$ times the distance between x and y, and centered at x and y, respectively.

Curve simplification

A general input reduction idea is the concept of curve simplification. It is often the case that the input curves contain too much redundant or irrelevant information, increasing the complexity of the curves, and thus of the problem. Then, given a curve, the problem is to find a simpler curve, i.e. of smaller complexity, while having a small dissimilarity to the original. We use the Fréchet distance as the dissimilarity measure, and develop an efficient simplification technique to capture critical points of the curves in one-dimensional space in Chapter 3.

Godau [86] noted that a simplification under the Fréchet distance is a bicriteria approximation. The parameters for the simplification of the curves can be either:

- the maximum allowed distance (error) between the original and the simplified curve, where the goal is to find a simplified curve with the minimum number of vertices within the given distance, or
- the maximum allowed complexity (size) of the simplified curve, and the goal is to minimize its distance to the original curve over all curves of the given size.

The problems can be observed with an additional constraint that the vertices of the simplified curves must belong to the original curve. In this case the simplification is called strong. Agarwal *et al.* [7] showed that the strong simplification is a 4-approximation to the unconstrained minimum-size problem. All of these problems are of independent research

interest, but they are useful as a building block for the other problems as well (e.g. in Chapter 4, or in [44]). For related work on the curve simplifications, see Subsection 3.1.3.

Through the curve simplification, we solve an additional problem – noise. For example, physical or chemical measurements typically have measurement errors. The stock and commodities market data contain minor transactions or short term trading, that (usually) do not affect the general trends (cf. [7]). By having a theoretical guarantee on the simplification quality, a better analysis of the input can be made.

Metric embeddings

The following approach to the problems on curves is a concept of metric embeddings and dimensionality reduction. In practice, the problems on multi-dimensional curves are often approximated by observing the coordinates separately. For example, for a stock market index, the performance of various stocks are combined. Frequently, a certain trend can be recognized by observing only one or a subset of stocks (cf. [7]).

A metric embedding is a mapping between two metric spaces which preserves distances up to a certain distortion. Any finite metric space with n points can be embedded isometrically into ℓ_∞^n [136], which is called the Fréchet embedding. Any bounded set in $\mathbb{R}^d$ with the metric induced by the ℓ_∞-norm can be embedded into the space of curves with the Fréchet distance (see Lemma 4.24). It is not known if the Fréchet distance spaces can be embedded into an ℓ_p-space using finite dimension [113]. The result of Bartal, Gottlieb and Neiman [24] implies that a metric embedding of the Fréchet distance into an ℓ_p-space must have at least super-constant distortion. For more results on metric embeddings and the Fréchet distance, see Subsection 5.1.3.

We are interested in the distortion of the embeddings of the Fréchet distance spaces into one-dimensional spaces (in Chapter 5). One such embedding was used by Sheehy [159] in his topological work on bounds of the Fréchet distance. His embedding picks a random point x in the original ambient space, and maps each vertex of the input curve to a point in $\mathbb{R}$ representing the distance of the vertex to x. However, no upper bound on such distortion for curves in $\mathbb{R}^d$ for arbitrary dimension d was given.

Bringmann and Künnemann [34] used projections to one-dimensional spaces (i.e. lines) to speed up their approximation algorithm for the Fréchet distance. They showed that the Fréchet distance computation can be done in linear time if the convex hulls of the two curves are disjoint, by reducing the decision problem to the one-dimensional separated curves.

A solution to the problem of metric embeddings under the Fréchet distance could serve as a building block for other fundamental computational tasks, such as clustering, nearest-neighbor search [69, 71], and spherical range search [4].

1.2.2 Clustering problems

Given a set of input objects (points, curves, etc.), it is a natural problem to look for a set of representatives under some given criteria, and/or to group the input objects according to that criteria, in order to extract patterns from the given set. The idea behind is that similar objects belong to the same group, and very dissimilar objects should be assigned to different groups. For example, bus stations should ideally be placed close to homes or working places in a city. Mountain huts should be evenly distributed and reachable to the walkers on various mountain trails. A criterion for grouping can also be negative, e.g. a diabetes warning campaign should not be placed together with a fast-food advertisement. A process of grouping the input objects is called *clustering*, and is defined using a cost function, that describes the pairwise distance between elements of the group. The groups are called *clusters*, and a representative of each group is called *cluster center*.

There is a plethora of various clustering problems in general metric spaces, and they have been extensively studied in many different settings. Let the maximum number of allowed clusters be denoted as k, $k \in \mathbb{N}$. We will discuss two intensively investigated clustering concepts that both have a goal to divide the input into k clusters, each with a cluster center that serves as a representative, thus we call them k-clustering problems. The first problem has the objective to find a set of k cluster centers such that the *maximum distance* to the nearest cluster center is minimized, over all input objects. This problem is known as the *k-center* clustering.

The goal of the second problem we consider is to find a set of k centers, that minimizes the *sum of distances* to the nearest centers, over all input objects. This problem is known as the *k-median* clustering. A popular problem related to the k-median problem, which we do not consider in this thesis, is the k-means clustering, where the distances are *squared* and then summed.

The 1-center problem in its simplest form with three input points in the plane is equivalent to the construction of the circumscribed circle to the triangle given by the three points. This problem and a compass-and-ruler construction were given by Euclid in his "*Elements*" (cf. [77] Book IV, Proposition 5) around year 300 BC. The 1-median problem with three points was posed by Fermat, followed by a compass-and-ruler construction by Toricelli (cf. [123]), both in 1600s.

When the number of clusters or the dimensionality of the problem is changed, the problems become very hard. It is known that the k-center and the k-median problem, both in general metric spaces and in Euclidean spaces, are **NP**-hard to be exactly solved, and in many cases, even to be approximated better than a constant factor (cf. [80, 101, 115, 138]). Several lines of approximation algorithms for the k-clustering problems exist, and we discuss some of them in Subsection 2.4.1. Some of these algorithms are limited only to the Euclidean spaces, others are intended for the general metric spaces. Sometimes, not all properties of metric spaces are provided, and this can actually be the case when we deal with the curves.

Ackermann, Blömer and Sohler [3] showed, that under certain conditions a randomized $(1+\varepsilon)$-approximation algorithm by Kumar, Sabharwal and Sen [126] can be extended to general distance measures. In particular, they showed that under a dissimilarity measure, one can compute a $(1+\varepsilon)$-approximation to the k-median problem if the 1-median problem solution can be $(1+\varepsilon)$-approximated, based only on information from a random sample of constant size, picked from the input. They showed that this is the case for metrics with bounded doubling dimension (cf. Definition 2.12), as it is the case for the Euclidean spaces. If the doubling dimension is unbounded, then the algorithm of [3] will remain applicable, provided that the dissimilarity measure satisfies certain conditions on sampling.

Clustering of curves

Clustering of curves is a fundamental problem of general interest: for industry to monitor the performance using sensor data [177], for statisticians to analyze functional [114, 156] or longitudinal data [55], for biologists to track the animals [94], for financial analysts to predict the market behavior [103], for data mining community [152], as well as for the computational geometry community. Unfortunately, most approaches are oriented to obtain a good *empirical* solution, and do not pursue *theoretical* guarantees on lower and upper bounds.

A direct approach to curve clustering under the Fréchet distance using k-clustering algorithms for general metric or Euclidean spaces fails on dimensionality of the ambient spaces. There is no known embedding into ℓ_p-spaces with finite dimension [113]. As previously said, the infinite doubling dimension of the space with Fréchet distance prevents the direct application of the standard techniques from [3, 126].

A generalization of the approach of Alt and Godau [14] for computing the Fréchet distance to k-clustering problems fails on dimensionality of the joint parametric space, defined by the set of the input curves. The algorithm of [14] explores the parametric space for a monotone and continuous mapping. If generalized to n curves, this approach leads to

the exponential time algorithms, as it does with a search for a single representative curve (cf. [12, 98]).

Furthermore, a representative curve under the Fréchet distance for an input of n curves of complexity m can have complexity of $O(mn)$ [12]. Such representative curves imply overfitted data, which is unnecessary for the real-world modelling. Therefore it is "natural" to introduce an additional parameter to the k-clustering problems for curves under the Fréchet distance, and to bound the complexity of the cluster centers by a constant ℓ. We formally introduce the (k, ℓ)-clustering problems in the one-dimensional ambient space in Subsection 4.1.1. It is straightforward to extend our definitions to curves in multi-dimensional ambient spaces.

It is important to assume that both k and ℓ are constants, since the (k, ℓ)-clustering problems are **NP**-hard, in the case when k is part of the input (cf. Section 4.6), and when ℓ is part of the input (cf. [44, 45]). For more related work on curve clustering, see Subsection 4.1.3. In this thesis we consider the (k, ℓ)-clustering problems for the curves in one-dimensional ambient space.

Clustering of probabilistic high-dimensional points

The k-clustering problems on probabilistic points, both for the general metric and Euclidean spaces were first studied by Cormode and McGregor [59]. They gave reductions of the probabilistic k-median and k-means problems to weighted instances of their corresponding deterministic problems. The probabilistic k-center problem was shown to be the most interesting one. All these probabilistic problems inherit the computational hardness from their deterministic counterparts. For more related work see Subsection 6.1.3.

In this thesis we consider the probabilistic version of the 1-center problem – the probabilistic smallest enclosing ball (pSEB), in $\mathbb{R}^d$. The (probabilistic) smallest enclosing ball problem is to find a center that minimizes the (expected) maximum distance to the input points. Munteanu, Sohler and Feldman [145] showed that the pSEB problem in $\mathbb{R}^d$ can be reduced to two deterministic instances of a generalized problem, that extends both 1-center and 1-median problem. For a formal definition and reduction details see Subsections 6.1.1 and 6.1.4.

The (probabilistic) SEB problem often occurs as a building block for complex data analysis and machine learning tasks like estimating the support of high dimensional distributions [169], outlier and novelty detection [166], anomaly detection [165], and classification and robot gathering [56]. It is therefore very important to develop highly efficient approximation algorithms for the base problem. This involves reducing the number of points but also keeping the dependence on the dimension as low as possible.

Munteanu, Sohler and Feldman [145] gave the first and fastest-to-date polynomial-time $(1+\varepsilon)$-approximation algorithm for the pSEB problem in fixed dimension. Its running time dependence on the number of points is linear, but exponential in the dimension. In particular, the number of realizations sampled by their algorithm had a linear dependence on dimension stemming from a ball-cover decomposition of the solution space. The actual algorithm made a brute force evaluation (on the sample) of all centers in a grid of exponential size in the dimension.

Our investigation is focused on the reduction of the dependence on the dimension, based on [145]. This is additionally motivated by the concept of kernel functions (cf. Subsection 2.1.2). Kernel methods are a standard technique in machine learning. These methods implicitly project the d-dimensional input data into much larger dimension D, that can even be infinite. However, in D-dimensional space, simple linear classifiers or spherical data fitting methods can be applied to obtain a non-linear separation or non-convex shapes in the original d-dimensional space. The efficiency of kernel methods is usually not affected. Despite the large dimension $D \gg d$, most important kernel functions, and thus inner products and distances in the D-dimensional space, can be evaluated in $O(d)$ time [155].

To make the probabilistic smallest enclosing ball algorithm viable in the context of kernel methods and generally in high dimensions, it is highly desirable to reduce the dependence on the dimension to a small polynomial occurring only in evaluations of inner products and distances between two (low dimensional) vectors.

1.3 Outline and contributions of this thesis

The rest of this thesis is structured into Chapters 2 – 6, dealing with the following content summarized below.

Chapter 2 presents the notions from, and the references on the body of the literature, that we need for analysis of the problems in the subsequent chapters. In particular, we give a brief overview of the metric and the normed spaces, and a sketch of the well-known subgradient descent method for convex optimization. We review the general metric and Euclidean k-center and k-median clustering problems and present the related work. Finally, we formally define the curves, that are the main objective of the present study, and the Fréchet distance to measure the dissimilarity of the curves. We close this chapter with a brief overview of the state-of-the-art on the computing of the Fréchet distance.

Each of the Chapters 3 – 6, is structured as follows. First we give an introduction followed by a formal definition of the problem, and an overview of the results presented in the

chapter. A discussion on the work related to the topic of the chapter is given afterwards. The rest of the chapter is dedicated to the analysis of the considered problem(s). Each chapter ends with a section discussing open problems and conclusions.

Chapter 3 is dedicated to the problem of **curve simplification**. We introduce a special type of curve simplification in the one-dimensional ambient space – the *δ-signatures*. We show several properties of signatures, and how to construct them efficiently, both in the case when the goal error and the goal size are given. Our signatures are designed in such way, that enables an efficient approach for other problems on curves in $\mathbb{R}$, e.g. to the clustering problems under the continuous Fréchet distance (in Chapter 4), and recently, for the first nearest neighbor constant-factor approximation algorithm under the continuous Fréchet distance [69].

In particular, we can use a technique similar to *shortcutting*, which has been used before in the context of partial curve matching (cf. [42, 65]). We show that given an input curve, any vertex of a clustering candidate center curve, that is not δ-close to a vertex of the δ-signature of the input curve, can be omitted from the candidate curve without increasing the distance beyond the threshold δ. This is the main technical contribution of Chapter 3, stated as Theorem 3.8.

Chapter 4 extends the classical k-center and k-median clustering problems to the (k, ℓ)-**curve clustering** problems, by adding a constraint on the number of vertices ℓ of the chosen representative curves. The input consists of n curves, each of complexity m. The dissimilarity measure of the curves is either the discrete or the continuous Fréchet distance. The curves in the one-dimensional ambient space are observed, and it is shown that even in such a setting, the observed problems are computationally hard. Under assumption that k, ℓ, and ε are constants, the first $(1 + \varepsilon)$-approximation algorithms for both (k, ℓ)-center and (k, ℓ)-median problems under the continuous Fréchet distance are given, in near-linear time in terms of the input. Our techniques also provide a constant-factor approximation algorithms for the (k, ℓ)-clustering problems under the discrete Fréchet distance.

Our approach exploits the low dimensionality of the ambient space, using signatures of bounded size. We show that the vertices of the potential cluster center curves need to be close to the vertices of the signatures of the input curves. This enables us to generate a constant-size set of candidate solutions for the (k, ℓ)-center problem.

In our analysis for the (k, ℓ)-median problem, we apply the known approach of random sampling of Kumar, Sabharwal and Sen [126] and Ackermann, Blömer and Sohler [3]. However, a straightforward application of their results is not possible, as we show that the

doubling dimension of the metric space of the curves under both discrete and continuous Fréchet distance is unbounded, even for the space of univariate curves.

We extend the conditions required for the application of the algorithm from [3] to take into account the added parameter ℓ, and show that one can generate a constant-size candidate set that contains a $(1+\varepsilon)$-approximation to the 1-median based on a sample of constant size (cf. Theorem 4.20). To achieve this, we observe that a vertex of the optimal solution, which is not close to a signature vertex of an input curve, and which is unlikely to be induced by our sample, can be omitted without increasing the cost by a factor of more than $(1+\varepsilon)$ (cf. Lemma 4.16).

Chapter 5 investigates the problem of the **embedding of the Fréchet distance** metric spaces. In particular, the embeddings into the space of one-dimensional curves are discussed, since for this space we have developed several efficient techniques described in the previous chapters. We ask for both upper and lower bounds on the distortion of a probabilistic embedding of the discrete Fréchet distance into the one-dimensional ambient space, under certain conditions on the dimension of the original ambient space. Some of our bounds extend to the continuous Fréchet distance, and to the dynamic time warping distance.

In our analysis we observe a *realistic* input class of polygonal curves – the c-packed curves. For this class of curves, with high probability, the distortion of such an embedding is shown to be linear in the complexity of the original curves (cf. Theorem 5.15). This may seem as a weak bound, but it is a first such result, and it is paired with the matching lower bound.

Chapter 6 contains a discussion on the **clustering of probabilistic data**. The objects of analysis of the clustering problems in this chapter are *probabilistic points*, contrary to the previous chapters that have dealt with curves. In particular, we are interested in solving an open problem posed by Munteanu, Sohler and Feldman [145]. They gave the first FPTAS to the probabilistic smallest enclosing ball (pSEB) problem, but under the assumption that the dimension of the ambient space is a constant. Based on algorithm from [145], we solve the problem for arbitrary dimension of the ambient space, by solving efficiently a generalized deterministic problem: the set median problem, extending the 1-center and 1-median problems.

Our solution for the set median problem is based on a popular method in convex optimization – the subgradient descent. We adapt the random sampling techniques of Kumar, Sabharwal and Sen [126] and Indyk and Thorup [110, 168] to avoid the dependence on the number of sets N, and keeping the dependence on the number of elements in each set n linear. Theorem 6.15 states the main technical contribution of Chapter 6. Additionally,

we show that the dependence on the number of elements in a set cannot be reduced to sublinear (after reading the input), unless we sacrifice the approximation factor of (roughly) $\sqrt{2}$, or accept the exponential dependence on the dimension (cf. Theorem 6.16).

Our probabilistic smallest enclosing ball $(1+\varepsilon)$-approximation algorithm requires running time that is no longer exponentially dependent on the dimension d of the ambient space, but only linear (cf. Theorem 6.17). Furthermore, the linear dependence on the dimension is needed only for the computation of the distances between the points. This enables us to further extend the pSEB algorithm to the probabilistic version of the support vector data description problem (SVDD). The SVDD problem is known to be equivalent to the smallest enclosing ball problem in (potentially infinitely) high dimensional feature space, induced by the kernel function. This result is stated as Theorem 6.19.

1.4 Coauthored sources

The present thesis is based on the previous joint work and the following publications, coauthored with Anne Driemel, Alexander Munteanu and Christian Sohler. The purpose of this section is to comply with the rules of good scientific practice at the TU Dortmund University [170]. For all publications with n authors, I have contributed $1/n$ of the work.

- Chapter 3 and Chapter 4 are based on [68],

  A. Driemel, A. Krivošija, and C. Sohler. Clustering time series under the Fréchet distance. In R. Krauthgamer, editor, *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 766–785, 2016

  The minor flaws are corrected. The argumentation is extended and partially rewritten to improve readability and to enhance the simplicity of verification. The differences for the discrete Fréchet distance case are explicitly discussed.

- Chapter 5 is based on [67],

  A. Driemel and A. Krivošija. Probabilistic embeddings of the Fréchet distance. In L. Epstein and T. Erlebach, editors, *16th International Workshop on Approximation and Online Algorithms, WAOA, Revised Selected Papers*, pages 218–237, 2018

  The minor errors are corrected. The argumentation is extended to improve readability. The upper bound is extended to the cases $d \in \{6, 7\}$ and discussed for the higher dimensions. The possibilities in the case of the continuous Fréchet distance are discussed.

- Chapter 6 is based on [125],

A. Krivošija and A. Munteanu. Probabilistic smallest enclosing ball in high dimensions via subgradient sampling. In G. Barequet and Y. Wang, editors, *Proceedings of the 35th International Symposium on Computational Geometry, SoCG*, volume 129 of *LIPIcs*, pages 47:1–47:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019

The argumentation is extended to improve readability and the simplicity of verification.

Some parts of the publications [67, 68, 125] are used and merged with other preliminaries in Chapter 2.

All related work, that was not coauthored by the author of this thesis, is listed and clearly denoted in the respective chapters.

2 Preliminaries

In this chapter we introduce the notions that are used throughout this thesis, and give references for further reading. We start with a brief overview of the notation used. Later, in Section 2.1 we make a brief review of metric spaces. In Sections 2.2 and 2.3 we state some basic results from the convex analysis and the probability theory. In Section 2.4 we introduce the clustering problems on general metric spaces and give an overview of the related work. We extend these problems to the space of one-dimensional curves in Chapter 4, and to a probabilistic generalization in Chapter 6. Finally, in Section 2.5 we define the curves and the Fréchet distance, and give an overview of the results on computing the versions of the Fréchet distance.

We denote the set of positive integers with $\mathbb{N}$, and the set of real numbers with $\mathbb{R}$. We denote the set of positive integers up to n with $[n] = \{1, \ldots, n\}$. For any set $\mathcal{P}$, we denote its cardinality with $|\mathcal{P}|$.

We use the following notational conventions:

- The calligraphic letters, e.g. $\mathcal{H}, \mathcal{S}$, are reserved for *sets and spaces*. Some sets are denoted with capital Latin letters (e.g. to distinct between a *set* and a *set of sets*).
- *Curves* are denoted with Greek letters $\sigma, \varsigma, \tau, \upsilon, \omega$.
- *Points* in $\mathbb{R}^d$ are denoted with small Latin letters, e.g. c, p, q, r.
- *Vectors* are denoted with bold letters, e.g. $\mathbf{u}, \mathbf{x}, \mathbf{y}$.
- Some Greek letters are used exclusively for *parameters*: γ for probabilities, δ for distances, and ε for approximation errors.

The following non-standard notation is used in this thesis.

- $\langle\langle a, b \rangle\rangle = [\min(a, b), \max(a, b)]$, for any $a, b \in \mathbb{R}$ (defined on page 47).
- $[h]_\delta = [h - \delta, h + \delta]$, for any $h \in \mathbb{R}$ and $\delta > 0$ (defined on page 47).
- $\mathbb{1}_E$: the indicator function of an event E (defined on page 29).
- $\mathbf{B}(p, r)$: the ball in a metric space, centered at p with radius r (defined on page 23).
- $d_{dF}(\cdot, \cdot)$: the discrete Fréchet distance (defined on page 40).
- $d_{DTW}(\cdot, \cdot)$: the dynamic time warping distance (defined on page 40).
- $d_F(\cdot, \cdot)$: the continuous Fréchet distance (defined on page 38).
- Δ: the set of polygonal curves in the ambient space $\mathbb{R}$ (defined on page 39).

- Δ_m: the set of polygonal curves of complexity m in the ambient space $\mathbb{R}$ (defined on page 39).
- $\mathcal{L}(\tau)$: the length of a curve τ (defined on page 37).
- $\max(\tau[t', t'']) = \max\{\tau(t) \colon t \in [t', t'']\}$, for a given curve $\tau \in \Delta$, and two parameters t' and t'', with $0 \leq t' \leq t'' \leq 1$. Analogously, we write $\min(\tau[t', t'']) = \min\{\tau(t) \colon t \in [t', t'']\}$ (defined on page 47).
- $\mathcal{V}(\tau)$: the set of the vertices of a curve τ (defined on page 36).

2.1 Notes on metric spaces

2.1.1 Metrics and norms

Given a set $\mathcal{X}$, a natural question to ask is to what extent are two elements $x, y \in \mathcal{X}$ (dis)similar to each other. To that end, any function $\mathbf{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty)$, such that for any $x, y \in \mathcal{X}$ it is $\mathbf{d}(x, y) = 0$ if and only if $x = y$, is called a **dissimilarity measure**. Additionally, we can require for any $x, y \in \mathcal{X}$, that it holds $\mathbf{d}(x, y) = \mathbf{d}(y, x)$. In that case, the function $\mathbf{d}$ is called a **distance function**. However, distance function properties are not sufficient for many problems, thus, an additional condition is added, introducing a notion of *metric*.

Definition 2.1 (**Metric, Pseudo-metric**, cf. [95] Definition 4.1)**.** *A metric space is a pair $(\mathcal{X}, \mathbf{d})$, where $\mathcal{X}$ is a set, and $\mathbf{d} : \mathcal{X} \times \mathcal{X} \to [0, \infty)$ is a function called metric, having the following properties for any three $x, y, z \in \mathcal{X}$:*

(i) $\mathbf{d}(x, y) = 0$ *if and only if* $x = y$ *(identity of indiscernible elements);*
(ii) $\mathbf{d}(x, y) = \mathbf{d}(y, x)$ *(symmetry);*
(iii) $\mathbf{d}(x, y) + \mathbf{d}(y, z) \geq \mathbf{d}(x, z)$ *(triangle inequality).*

A function $\mathbf{d}$ is a pseudo-metric, if all aforementioned properties are satisfied, except possibly (i).

Throughout this thesis we will work with the space $\mathbb{R}^d$, $d \in \mathbb{N}$, whose elements we call **points**. For $x \in \mathbb{R}^d$, we denote its coordinates with x_i, for $i \in [d]$. Then, the Euclidean distance between two d-dimensional points x and y is defined by

$$\mathbf{d}(x, y) = \left(\sum_{i=1}^{d} |x_i - y_i|^2 \right)^{1/2}. \tag{2.1}$$

The set $\mathbb{R}^d$ accompanied with the Euclidean distance (Equation (2.1)) is a metric space (cf. [95, 158]).

A **vector space** $\mathcal{V}$ over a field F is a set of objects $\mathcal{V}$ (called *vectors*) that is closed under addition operation, that is associative and commutative and has an identity (known as *zero vector*). The set $\mathcal{V}$ contains additive inverses and is closed under multiplication with scalars from the field F (cf. [102], Section 0.1). Throughout this work we use the field $F = \mathbb{R}$.

The set $\mathbb{R}^d$, $d \in \mathbb{N}$, is a vector space over the field $\mathbb{R}$ (cf. [102]). We call it Euclidean d-dimensional space. The coordinates of a vector $\mathbf{x} \in \mathbb{R}^d$ are denoted with x_i, $i \in [d]$, and write $\mathbf{x} = (x_1, \ldots, x_d)$. A point $x \in \mathbb{R}^d$ is associated with its position vector $\mathbf{x} \in \mathbb{R}^d$, thus we can overload the coordinates' notation. In the geometric interpretation, the vector $\mathbf{x}$ begins at origin in $\mathbb{R}^d$, and ends at the point x.

A norm intuitively introduces a concept of length of the elements of a vector space.

Definition 2.2 (**Norm**, cf. [102] Definition 5.1.1)**.** *Let $\mathcal{V}$ be a vector space over the field F. A function $\|\cdot\|\colon \mathcal{V} \to \mathbb{R}$ is a norm, if, for all $\mathbf{x}, \mathbf{y} \in \mathcal{V}$ and all $\alpha \in F$:*

(i) $\|\mathbf{x}\| \geq 0$ *(nonnegativity);*

(ii) $\|\mathbf{x}\| = 0$ *if and only if* $\boldsymbol{x} = 0$ *(positivity);*

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|$ *(homogeneity);*

(iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ *(triangle inequality).*

Sometimes the notion *vector norm* is used instead. In many aspects it resembles the definition of the metric. Indeed, any norm $\|\cdot\|$ defines a metric $\mathbf{d}$ using $\mathbf{d}(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|$, but the opposite does not hold (cf. [158]). The vector space equipped with a norm is called a **normed space**.

We use the ℓ_p-norms, defined on the vector space $\mathcal{V} = \mathbb{R}^d$ over the field $\mathbb{R}$ (cf. [102] Section 5.2). The ℓ_p-norm of $\mathbf{x} \in \mathbb{R}^d$, denoted $\|\mathbf{x}\|_p$, for $p \in [1, \infty)$ is defined as

$$\|\mathbf{x}\|_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{1/p}. \tag{2.2}$$

The limiting case $p = \infty$ is defined as $\|\mathbf{x}\|_\infty = \max_{i \in [d]} |x_i|$. All ℓ_p-norms satisfy the conditions of Definition 2.2 (cf. [157]). The normed space $(\mathbb{R}^d, \|\cdot\|_p)$ is called the ℓ_p-space, and compactly denoted with ℓ_p^d.

The ℓ_1-norm is known as the sum, Manhattan, or taxicab norm. The ℓ_2-norm is the Euclidean norm. The ℓ_∞-norm is called the maximum norm. Throughout this thesis we mostly use the Euclidean norm, and therefore, for simplicity of notation, we write $\|\cdot\|$ instead of $\|\cdot\|_2$. We emphasize the type of the ℓ_p-norm we use only in the case that multiple norms are discussed in one context.

We note that the Euclidean distance between two points $x, y \in \mathbb{R}^d$ equals the Euclidean norm of the vector $\mathbf{x} - \mathbf{y}$, defined by their respective position vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Therefore, it is sometimes convenient to omit the vector notation, and denote the Euclidean distance metric (Equation 2.1) between the points x and y with the norm $\|x - y\|$, as long as the distinction is clear from the context.

2.1.2 Inner products, Hilbert spaces, Kernel functions

After introducing the concepts of distances and lengths, we need the concept of *angles*. This is obtained through the notion of the *inner products* on the normed spaces.

Definition 2.3 (**Inner product**, cf. [102] Definition 5.1.3)**.** *Let $\mathcal{V}$ be a vector space over the field F. A function $\langle \cdot, \cdot \rangle : \mathcal{V} \times \mathcal{V} \to F$ is an inner product if for all $\mathbf{x}, \mathbf{y}, \mathbf{z} \in \mathcal{V}$ and all $\alpha \in F$,*

(i)	$\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$	*(nonnegativity);*
(ii)	$\langle \mathbf{x}, \mathbf{x} \rangle = 0$ *if and only if* $\mathbf{x} = 0$	*(positivity);*
(iii)	$\langle \mathbf{x} + \mathbf{y}, \mathbf{z} \rangle = \langle \mathbf{x}, \mathbf{z} \rangle + \langle \mathbf{y}, \mathbf{z} \rangle$	*(additivity);*
(iv)	$\langle \alpha \mathbf{x}, \mathbf{y} \rangle = \alpha \langle \mathbf{x}, \mathbf{y} \rangle$	*(homogeneity);*
(v)	$\langle \mathbf{x}, \mathbf{y} \rangle = \overline{\langle \mathbf{y}, \mathbf{x} \rangle}$	*(Hermitian property).*

Since we only use the field $\mathbb{R}$, the Hermitian property in Definition 2.3 becomes the symmetry property. A vector space with inner product is called an **inner product space**. Any inner product $\langle \cdot, \cdot \rangle$ on a vector space $\mathcal{V}$ defines a norm $\|\cdot\|$ on $\mathcal{V}$, using $\|x\| = \langle x, x \rangle^{1/2}$, but the opposite does not hold (cf. [158]). Since every norm defines a metric, any inner product space is a metric space.

On the Euclidean vector space $(\mathbb{R}^d, \|\cdot\|_2)$ the inner product, denoted $\langle \cdot, \cdot \rangle_2$, is defined for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ as

$$\langle \mathbf{x}, \mathbf{y} \rangle_2 = \mathbf{x}^T \mathbf{y} = \sum_{i=1}^{d} x_i y_i. \tag{2.3}$$

It is clear from Equation (2.3) that for every $\mathbf{x} \in \mathbb{R}^d$, it is $\langle \mathbf{x}, \mathbf{x} \rangle_2 = \|\mathbf{x}\|_2^2$. The Euclidean inner product on the vector space $\mathbb{R}^d$ is the one we mostly use, thus we overload the notation and simply write $\langle \cdot, \cdot \rangle$ instead of $\langle \cdot, \cdot \rangle_2$. The only exception will be in the context of kernel functions, where the type of the inner product will be duly noted.

The Cauchy-Schwarz inequality is an important property of all inner products. We state it here for the Euclidean space $\mathbb{R}^d$.

Lemma 2.4 (**Cauchy-Schwarz inequality**, cf. [102] Theorem 5.1.4)**.** *Let $\langle \cdot, \cdot \rangle$ be the inner product on a Euclidean vector space $\mathbb{R}^d$. Then it holds that*

$$|\langle \mathbf{x}, \mathbf{y} \rangle|^2 \leq \langle \mathbf{x}, \mathbf{x} \rangle \langle \mathbf{y}, \mathbf{y} \rangle \quad \text{for all} \quad \mathbf{x}, \mathbf{y} \in \mathbb{R}^d, \tag{2.4}$$

with equality if and only if $\mathbf{x}$ and $\mathbf{y}$ are linearly dependent, that is, if and only if $\mathbf{x} = \alpha \mathbf{y}$ for some $\alpha \in \mathbb{R}$.

The geometric interpretation of the inner product in $\mathbb{R}^d$ is that it defines a cosine of the angle between vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$. Let the angle φ be the angle between vectors $\mathbf{x}$ and $\mathbf{y}$, i.e. $\varphi = \angle x0y$. Then

$$\cos \varphi = \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}, \tag{2.5}$$

for $\varphi \in [0, \pi]$ (cf. [158]).

Sometimes the inner product spaces are called *pre-Hilbert spaces*. For the definition of Hilbert spaces, that we will need for an application of the results in Chapter 6, we need first to define the Cauchy sequence.

Definition 2.5 (**Cauchy sequence**, cf. [158] Definition B.10)**.** *A sequence $\{\mathbf{x}_i\}_{i \in \mathbb{N}}$ in a normed space $\mathcal{H}$ is said to be a Cauchy sequence, if for every $\varepsilon > 0$, there exists an $n \in \mathbb{N}$, such that for all $n', n'' > n$ it is $\|\mathbf{x}_{n'} - \mathbf{x}_{n''}\| < \varepsilon$. A Cauchy sequence converges to an element $\mathbf{x} \in \mathcal{H}$ if $\|\mathbf{x}_n - \mathbf{x}\| \to 0$ as $n \to \infty$.*

Definition 2.6 (**Banach and Hilbert spaces**, cf. [158] Definition B.11)**.** *A space $\mathcal{H}$ is called complete if all Cauchy sequences in the space converge. A Banach space is a complete normed space. A Hilbert space is a complete inner product space.*

The Euclidean space $\mathbb{R}^d$ is the simplest Hilbert space. However, the dimension of the Hilbert spaces can be unbounded [158].

To introduce the SVDD problem in Chapter 6, we need to introduce the notion of *kernel functions*. Kernel functions can be defined on any set $\mathcal{X}$. In this thesis, we use $\mathcal{X} = \mathbb{R}^d$. A positive semidefinite function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ is called a **kernel function** (sometimes called *covariance function*, cf. [155] Section 4.1).

It is well known by Mercer's theorem (cf. [155] Theorem 4.2, or [158] Theorem 2.10), that such a function implicitly defines the inner product, denoted $\langle \cdot, \cdot \rangle_{\mathcal{H}}$, in a high dimensional Hilbert space $\mathcal{H}$, say $\mathbb{R}^D$, where $D \gg d$.

This means we have $K(\mathbf{x}, \mathbf{y}) = \langle \phi(\mathbf{x}), \phi(\mathbf{y}) \rangle_{\mathcal{H}}$, for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, where $\phi \colon \mathbb{R}^d \to \mathcal{H}$ is the so-called **feature mapping** associated with the kernel. We call the space $\mathcal{H}$ associated with the kernel K its **feature space**. The idea is to be able to work implicitly in the

high-dimensional space $\mathcal{H}$ by interaction only through the inner product, and to replace the inner product by an invocation of the kernel function K. Then, we could act as if the function ϕ was computed explicitly, without actually computing ϕ (cf. [28] Section 5.3). This is sometimes called the kernel trick.

Examples for such kernel functions include polynomial transformations of the standard inner product in $\mathbb{R}^d$ such as the constant, linear or higher order polynomial kernels, e.g. $K(\mathbf{x}, \mathbf{y}) = \text{poly}(\langle \mathbf{x}, \mathbf{y} \rangle)$. In these cases the dimension D (of the Hilbert space) remains bounded, but grows as a function of d raised to the power of the polynomials' degree.

Other examples are the exponential, squared exponential, Matérn, or rational quadratic kernels, which are transformations of the Euclidean distance between the two low dimensional vectors. The dimension D of the implicit feature space associated with these kernels is in principle unbounded, e.g. for the exponential kernel function $K(\mathbf{x}, \mathbf{y}) = e^{-\|\mathbf{x}-\mathbf{y}\|^2}$ (cf. [155], Table 4.1). Despite the large dimension $D \gg d$, all these kernels can be evaluated in time $O(d)$ (cf. [155]). For a longer introduction to kernel functions and their applications we confer to the books by Rasmussen and Williams [155] and by Schölkopf and Smola [158].

2.1.3 Embeddings of the metric spaces

A metric embedding is a function between two metric spaces which preserves the distances between the elements of the metric space. Definition 2.7 provides a formal definition of metric spaces.

Definition 2.7 (**D-embedding of metric spaces, Distortion**, cf. [136] Definition 15.1.1)**.** *Given metric spaces $(\mathcal{X}, \mathbf{d}_X)$ and $(\mathcal{Y}, \mathbf{d}_Y)$, a mapping $f : \mathcal{X} \to \mathcal{Y}$ is called a D-embedding, where $D \geq 1$ is a real number, if there exists a number $r > 0$ such that for all $\mathbf{u}, \mathbf{v} \in \mathcal{X}$,*

$$r \cdot \mathbf{d}_X(\mathbf{u}, \mathbf{v}) \leq \mathbf{d}_Y(f(\mathbf{u}), f(\mathbf{v})) \leq D \cdot r \cdot \mathbf{d}_X(\mathbf{u}, \mathbf{v}).$$

The infimum of the numbers D such that f is a D-embedding is called the distortion of f.

The embedding is called **isometric** if distances are preserved exactly, i.e. if the distortion of the embedding is 1.

In Section 4.6 we present an embedding necessary for our hardness result. We discuss the distortion of embeddings of the metric spaces of polygonal curves in Chapter 5. Moreover, for a discussion in Section 2.4 we need a following well-known result on isometric embedding between spaces ℓ_1^d and $\ell_\infty^{2^{d-1}}$, for all $d \in \mathbb{N}$. For $d = 2$, Lemma 2.8 implies that ℓ_1^2 is isometrically embeddable into ℓ_∞^2.

Lemma 2.8 (cf. [111]). *For any $d \in \mathbb{N}$, the space $\mathbb{R}^d$ with the ℓ_1-norm is isometrically embeddable into the space $\mathbb{R}^{2^{d-1}}$ with the ℓ_∞-norm.*[2]

2.1.4 Balls, spheres, and doubling dimension

In a metric space $\mathcal{X}$ it is often important to identify all elements of $\mathcal{X}$ that are at most at certain distance from a given element in $\mathcal{X}$. This is captured by the notion of *balls*.

Definition 2.9 (**Ball**). *In a metric space $(\mathcal{X}, \mathbf{d})$, a ball of center $p \in \mathcal{X}$ and radius $r \in \mathbb{R}$, $r \geq 0$, is defined as the set $\mathbf{B}(p, r) = \{q \in \mathcal{X} \colon \ \mathbf{d}(p, q) \leq r\}$.*

Wherever the term "ball" is used throughout this thesis it will be clear from the context which metric space is used. Therefore, we do not distinguish the metric space from the notation of a ball.

In the Euclidean spaces, a d-sphere is a d-dimensional manifold that can be embedded in Euclidean $(d+1)$-dimensional space. A d-sphere is a set of points in $\mathbb{R}^{d+1}$ at a constant distance r from a fixed point called center. A ball in Euclidean d-dimensional space is called a d-ball, and it is bounded by a $(d-1)$-sphere. The d-sphere is the surface or boundary of a $(d+1)$-dimensional ball.

We want to be able to quantify the volume and the surface area of d-spheres with radius r in the Euclidean space. To do so, we need first to introduce the *gamma function*. It is a known extension of the factorial function to the set of complex numbers, but here we give the definition using the set of positive real numbers as a domain.

Definition 2.10 (**Gamma function**, cf. [17] Equation 5.2.1). *Given positive real number z, the gamma function is defined by*

$$\Gamma(z) = \int_0^\infty x^{z-1} e^{-x} dx.$$

The gamma function has the following properties. It holds that $\Gamma(1) = 1$ and $\Gamma(n+1) = n!$ for all $n \in \mathbb{N}$ ([17] Equation 5.4.1). It is $\Gamma(1/2) = \sqrt{\pi}$ ([17] Equation 5.4.6). Since for all $z > 0$ it is $\Gamma(z+1) = z \cdot \Gamma(z)$ ([17] Equation 5.5.1), we can conclude by induction that, for all $n \in \mathbb{N} \cup \{0\}$, it is

$$\Gamma\left(n + \frac{1}{2}\right) = \frac{(2n)!\sqrt{\pi}}{4^n \cdot n!}.$$

[2] In the survey by Indyk [111] the goal dimension is 2^d. The result with 2^{d-1} is proven in the lecture notes [23] as Theorem 1.14. See `https://moodle2.cs.huji.ac.il/nu15/pluginfile.php/553935/mod_resource/content/1/METAP18_Lecture_1.pdf` (visited on April 13th, 2021).

Now, the volume and the surface area of a d-dimensional Euclidean ball are given by the following lemma.

Lemma 2.11 (cf. [17] Equation 5.19.4)**.** *The volume $V_d(r)$ and the surface area $S_d(r)$ of the d-dimensional ball ($(d-1)$-sphere) of radius $r \geq 0$ in the space $(\mathbb{R}^d, \|\cdot\|_2)$ are given by*

$$V_d(r) = \frac{\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}+1\right)} \cdot r^d \qquad \text{and} \qquad S_d(r) = \frac{2\pi^{\frac{d}{2}}}{\Gamma\left(\frac{d}{2}\right)} \cdot r^{d-1}. \tag{2.6}$$

The concept of *doubling dimension* of a metric space seems to be primarily of combinatorial interest, using balls in that metric space. However, it turns out that, if the doubling dimension is finite, it provides properties to the metric space that enable efficient clustering algorithms, discussed in Section 2.4.

Definition 2.12 (**Doubling dimension**, cf. [93])**.** *The doubling dimension of a metric space is the smallest positive integer d such that every ball of the metric space can be covered by 2^d balls of half the radius.*

Note that for every fixed $p \geq 1$, the doubling dimension of the space $(\mathbb{R}^d, \|\cdot\|_p)$ is $\Theta(d)$ (cf. the work of Gupta, Krauthgamer and Lee [93]). In many applications, the intrinsic dimension of the input data is much lower than the dimension of the space the data is taken from (cf. Ackermann, Blömer and Sohler [3]). However, the opposite can also be the case, as we show in Section 4.2, where the ambient space has a finite doubling dimension, but the space of the curves has an infinite doubling dimension.

2.1.5 Notes from the real analysis

In this thesis, we use the following two concepts from the real analysis. First concept provides a characterization of mappings between two metric spaces. The Lipschitz continuous property, stated by Definition 2.13, is stronger than the uniform continuity property, but weaker than the continuous differentiability property (cf. [157]). This turns out to be a sufficient property for the subgradient optimization method to work (cf. Section 2.2), and later for its probabilistic extension in Chapter 6, as the functions we optimize are not differentiable.

Definition 2.13 (**Lipschitz function**, cf. [157] on page 192)**.** *A mapping f from a metric space $(\mathcal{X}, \mathbf{d}_X)$ to a metric space $(\mathcal{Y}, \mathbf{d}_Y)$ is said to be M-Lipschitz continuous, provided there is a constant $M \geq 0$ such that for all $u, v \in \mathcal{X}$ it is*

$$\mathbf{d}_Y(f(u), f(v)) \leq M \cdot \mathbf{d}_X(u, v). \tag{2.7}$$

Sometimes the Lipschitz property (Equation (2.7)) is valid only within a ball $\mathbf{B}(c, R) \subset \mathcal{X}$, with given center $c \in \mathcal{X}$ and radius $R > 0$. Then we say that f is M-Lipschitz on $\mathbf{B}(c, R)$.

The second concept is the *Lebesgue measure*. It is convenient to use the Lebesgue measure to simplify the expressions including total length of intervals contained in the finite or countable collection, in the one-dimensional space $\mathbb{R}$.

For a set of real numbers $\mathcal{U} \subseteq \mathbb{R}$, we denote its **Lebesgue measure** with $\mu(\mathcal{U})$ (cf. the book by Royden [157], Chapter 2 for more details). This set function has following three properties:

i) each interval $I = [a, b]$ is measurable, and its measure $\mu(I)$ is its length, which is defined to be $|b - a|$ if I is bounded, and ∞ if I is unbounded;
ii) measure is transition invariant, i.e. for a measurable set E and $x \in \mathbb{R}$, if we define $E + x = \{e + x \colon e \in E\}$, then it is $\mu(E + x) = \mu(x)$;
iii) measure is countably additive over countable disjoint unions of sets, i.e. if $\{E_i\}_{i=1}^{\infty}$ is a countable pairwise disjoint collection of measurable sets, then $\mu(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mu(E_i)$.

Since for any two measurable sets E_1 and E_2 in $\mathbb{R}$ it is $\mu(E_1 \cup E_2) + \mu(E_1 \cap E_2) = \mu(E_1) + \mu(E_2)$ (cf. [157] page 47), it follows that for a finite set $\mathcal{I}$ of n (possibly not pairwise disjoint) intervals I_i, $i \in [n]$, the value of $\mu(\mathcal{I}) = \mu\left(\cup_{i=1}^{n} I_i\right)$ is the total length of all intervals in $\mathcal{I}$.

2.2 Notes from the convex analysis

In Chapter 6 we use several results from the convex analysis, that are needed for the subgradient descent method. In this section, we list these results. For a longer introduction to convex optimization confer to the books by Boyd and Vanderberghe [29], and by Nesterov [148]. We start with the notion of a convex set.

Definition 2.14 (**Convex set**, cf. [148] Definition 2.2.2)**.** *A set $\mathcal{Q} \subseteq \mathbb{R}^d$ is called convex, if for any $x, y \in \mathcal{Q}$, their assigned vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, and any $\lambda \in [0, 1]$, it holds that the point z, representing the vector $\mathbf{z} = \lambda\mathbf{x} + (1 - \lambda)\mathbf{y}$, is in $\mathcal{Q}$.*

The point z is called a convex combination of x and y. We can reduce the notation to the points only, implying the assigned vectors. It is well known (cf. [29] Section 2.1.4), that for any given set of points $\mathcal{Q} = \{x_1, \ldots, x_n\}$ in $\mathbb{R}^d$, its **convex hull**, denoted $CH(\mathcal{Q})$, and defined as the set of all convex combinations of the points in $\mathcal{Q}$:

$$CH(\mathcal{Q}) = \{y \in \mathbb{R}^d \colon y = \sum_{i=1}^{n} \lambda_i x_i, \text{ where } x_i \in \mathcal{Q}, \lambda_i \geq 0, \sum_{i=1}^{n} \lambda_i = 1, \text{ for all } i \in [n]\} \quad (2.8)$$

is a convex set.

Definition 2.15 (**Convex function**, cf. [29] Equation 1.3)**.** *A function $\phi : \mathbb{R}^d \to \mathbb{R}$ is called a convex function, if for all $x, y \in \mathbb{R}^d$, and for all $\alpha, \beta \geq 0$ with $\alpha + \beta = 1$, it holds that*

$$\phi(\alpha x + \beta y) \leq \alpha\phi(x) + \beta\phi(y). \quad (2.9)$$

The link between convex sets and convex functions is made by the epigraph: a function is convex if and only if its epigraph, defined as $\text{epi}(\phi) = \{(x, t) \in \mathcal{D} \times \mathbb{R} \colon \phi(x) \leq t\}$, is a convex set (cf. [29] Section 3.1.7).

One of the main problems in convex optimization is to solve

$$\min\{\phi(x) \colon x \in Q\}, \quad (2.10)$$

where ϕ is a convex function, and Q is a closed convex set $Q \subseteq \mathbb{R}^d$ (cf. [148] Equation 3.2.7). If the gradients of ϕ exist, i.e. ϕ is continuously differentiable, then there are many methods to solve the optimization problem. However, the differentiability of the function ϕ is often not provided, and therefore, the gradients do not exist. For such general convex functions, the notion of a subgradient is appropriate.

Definition 2.16 (**Subgradient, subdifferential**, cf. [148] Definition 3.1.6)**.** *Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be a convex function. A vector $\mathbf{g}$ is called a subgradient of function ϕ at point x, if for any $y \in \mathbb{R}^d$ it holds*

$$\phi(x) - \phi(y) \leq \langle \mathbf{g}, x - y \rangle . \quad (2.11)$$

The set of all subgradients of ϕ at x, denoted $\partial\phi(x)$, is called the subdifferential of function ϕ at point x.

The set of subgradients has many useful properties. For more details we refer to a book by Nesterov [148], and state here the following two, as Lemma 2.17 and Lemma 2.18. A convex function $\phi : \mathcal{D} \to R$, $\mathcal{D} \subseteq \mathbb{R}^d$, is **closed**, if its epigraph is closed.

Lemma 2.17 (cf. [148] Theorem 3.1.13)**.** *Let ϕ be a closed and convex function, and let x_0 be a point in the interior of the domain of ϕ. Then, $\partial\phi(x_0)$ is a non-empty set.*

Since the domain of the function ϕ will be $\mathbb{R}^d$ for our applications, we can simply use the fact that the subdifferential is always non-empty. If the zero-vector belongs to the subdifferential of the function ϕ in some point, then that point is a (local) optimum of ϕ.

Lemma 2.18 (cf. [148] Theorem 3.1.15)**.** *Given function[3] $\phi : \mathcal{D} \to \mathbb{R}$, $\mathcal{D} \subseteq \mathbb{R}^d$, we have that $\phi(x^*) = \min_{x \in \mathcal{D}} \phi(x)$ if and only if $\mathbf{0} \in \partial\phi(x^*)$.*

The convexity of a function $\phi : \mathbb{R}^d \to \mathbb{R}$ implies that any local optimum of ϕ is also the global optimum. We state this as the next lemma, followed by a standard proof.

Lemma 2.19. *Let $\phi : \mathcal{D} \to \mathbb{R}$, $\mathcal{D} \subseteq \mathbb{R}^d$, be a convex function, and let x^* be a local minimum of ϕ. Then it is $f(x^*) \leq f(x)$, for all $x \in \mathcal{D}$.*

Proof. Since x^* is a local minimum, it holds for an arbitrary small $\varepsilon > 0$, for all $x \in \mathcal{D}$ with $\|x - x^*\| \leq \varepsilon$, that $f(x) \geq f(x^*)$. Let there exist another point $\hat{x}$, that is a global minimum of ϕ, with $\hat{x} \neq x^*$, i.e. $\phi(\hat{x}) < \phi(x^*)$. By convexity of ϕ, for all $\alpha \in (0, 1]$ it is

$$\phi(\alpha\hat{x} + (1-\alpha)x^*) \leq \alpha\phi(\hat{x}) + (1-\alpha)\phi(x^*) < \alpha\phi(x^*) + (1-\alpha)\phi(x^*) = \phi(x^*). \quad (2.12)$$

Let arbitrarily small $\varepsilon > 0$ be chosen by the local minimum property. Let $x_\alpha = \alpha\hat{x} + (1-\alpha)x^* \in \mathcal{D}$. Then it is

$$\|x_\alpha - x^*\| = \|(\alpha\hat{x} + (1-\alpha)x^*) - x^*\| = \|\alpha\hat{x} - \alpha x^*\| = \alpha \|\hat{x} - x^*\|. \quad (2.13)$$

If $\|\hat{x} - x^*\| < \varepsilon$, then it suffices to choose $\alpha = 1$. Otherwise, let $\alpha = \varepsilon / \|\hat{x} - x^*\| \leq 1$. For the chosen value of α, it follows from Equation (2.13) that $\|x_\alpha - x^*\| \leq \varepsilon$. But then Equation (2.12) contradicts the assumption that x^* is a local minimum. □

The **subgradient descent optimization method** to solve the problem of Equation (2.10) consists of a simple iterative scheme, that we describe next (as presented in the book by Nesterov [148] Section 3.2.3). We start from some point $c_0 \in Q$. Let $\{h_i\}_{i \in \mathbb{N} \cup \{0\}}$ be a sequence, such that $h_i > 0$, $\lim_{i \to \infty} h_i = 0$, and $\sum_{i=0}^{\infty} h_i = \infty$. The

[3] The statement of this lemma does not require for the function ϕ to be convex. However, the subdifferentiability of the function in x^* implies convexity [148].

point c_{i+1} is computed in the i-th iteration. Let $g(c_i)$ be a subgradient of ϕ at point c_i. Then, let $c_{i+1} = c_i - h_i \cdot g(c_i)/\left\|g(c_i)\right\|$, i.e. the point c_i is translated by the unit vector $g(c_i)/\left\|g(c_i)\right\|$, multiplied by $-h_i$. Provided that ϕ is M-Lipschitz continuous for some constant M (cf. Definition 2.13), this method converges toward the optimal solution point. This is stated by the following theorem.

Theorem 2.20 (**Subgradient descent**, cf. [148] Theorem 3.2.2)**.** *Let $c^* \in \operatorname{argmin}_{c\in\mathbb{R}^d} \phi(c)$, i.e. c^* be an optimal solution of the problem of Equation (2.10). Let c_0 be the chosen starting point. Let ϕ be M-Lipschitz continuous on $\mathbf{B}(c^*, R)$, where $R = \|c_0 - c^*\|$. Then after i iterations of the subgradient descent method, it holds that*

$$\min_{j\in\{0,\ldots,i\}} \phi(c_j) - \phi\left(c^*\right) \leq M \cdot \frac{R^2 + \sum_{j=0}^{i} h_j^2}{2\sum_{j=0}^{i} h_j}. \tag{2.14}$$

The subgradient descent method converges toward an optimal solution. However, we can obtain a guaranteed quality of the solution after a fixed number of iterations, say ℓ, by setting $h_i = R/\sqrt{\ell+1}$, for $i \in \{0, 1, \ldots, \ell\}$. Then, Equation (2.14) becomes

$$\min_{j\in\{0,\ldots,\ell\}} \phi(c_j) - \phi\left(c^*\right) \leq \frac{MR}{\sqrt{\ell+1}}. \tag{2.15}$$

This strategy is also optimal for the problem of Equation (2.10) (cf. [148]).

2.3 Results from the probability theory

A **probability space** is a triple $(\Omega, \mathcal{F}, \mathbf{Pr})$, containing:

- a sample space Ω, which is the set of all possible outcomes of the random process modeled by the probability space;
- a family $\mathcal{F}$ representing allowable events, and each set in $\mathcal{F}$ is a subset in Ω; and
- a probability function $\mathbf{Pr} : \mathcal{F} \to \mathbb{R}$, that satisfies Definition 2.21.

An element of Ω is called a simple event. The definition of probability spaces is taken from [141] Definition 1.1.

Definition 2.21 (**Probability function**, cf. [141] Definition 1.2)**.** *A probability function is any function $\mathbf{Pr} : \mathcal{F} \to \mathbb{R}$, that satisfies the following conditions:*

(i) for any event $E \in \mathcal{F}$ it is $0 \leq \mathbf{Pr}\left[E\right] \leq 1$;

(ii) $\mathbf{Pr}\left[\Omega\right] = 1$; and

(iii) for any finite or countably infinite sequence of pairwise mutually disjoint events $\{E_i\}_{i\geq 1}$ it is $\mathbf{Pr}\left[\cup_{i\geq 1} E_i\right] = \sum_{i\geq 1} \mathbf{Pr}\left[E_i\right]$.

The probability spaces that are used throughout this thesis are discrete probability spaces, i.e. the sample space Ω is finite or countably infinite, and the family $\mathcal{F}$ consists of all subsets of Ω. We use the following notation. For any event E let the indicator function be $\mathbb{1}_E = 1$ if E happens, and 0 otherwise. The probability of an event $E \in \mathcal{F}$ is denoted $\mathbf{Pr}[E]$. For a random variable X and a probability distribution D, we write $X \sim D$ to indicate that X is distributed according to D. The expected value of a random variable $X \sim D$ over a sample space Ω is denoted as $\mathbb{E}[X] = \sum_{x \in \Omega} x \cdot \mathbf{Pr}[X = x]$. If we have a function ϕ that depends on the randomness of two random variables X and Y, say $\phi(X, Y)$, then we write $\mathbb{E}_X[\phi(X, Y)]$ to emphasize that the expectation is only over randomness of X.

In the rest of this section we state some well-known results from the probability theory, that are used throughout this thesis. For more results and a well-written introduction to randomness-based algorithms see the book by Mitzenmacher and Upfal [141].

Theorem 2.22 (**Linearity of expectation**, cf. [141] Theorem 2.1)**.** *For any finite collection of discrete random variables $X_1, \ldots, X_n$ with finite expected values, it is*

$$\mathbb{E}\left[\sum_{i=1}^{n} X_i\right] = \sum_{i=1}^{n} \mathbb{E}[X_i].$$

If two random variables X and Y are given on the same probability space, then the expression $\mathbb{E}[X \mid Y]$ is a random variable $f(Z)$, i.e. a function of the random variable Z, that takes the value $\mathbb{E}[X \mid Y = y]$ when $Y = y$ (cf. [141] Definition 2.7). Then, from the linearity of expectation, we have the following lemma.

Lemma 2.23 (**Law of total expectation**, cf. [141] Theorem 2.7)**.** *Let X and Y be two random variables on the same probability space. Then it holds that*

$$\mathbb{E}_X[X] = \mathbb{E}_Y[\mathbb{E}_X[X|Y]]. \tag{2.16}$$

The geometric random variable represents the situation where we have a sequence of independent trials, and in each trial the success probability equals p.

Definition 2.24 (**Geometric random variable**, cf. [141] Definition 2.8)**.** *A geometric random variable X with parameter p is given by the following probability distribution on $n \in \mathbb{N}$:*

$$\mathbf{Pr}[X = n] = (1 - p)^{n-1} p. \tag{2.17}$$

That is, for the geometric random variable X to equal n, there must be $n-1$ failures. For a random variable X with the distribution given by Equation (2.17), the expectation is given by

$$\mathbb{E}[X] = \frac{1}{p}. \tag{2.18}$$

A simple but very useful fact, presented by the following lemma, intuitively states that the probability that an event from a collection of events occurs is at most the sum of probabilities of single events from that collection.

Lemma 2.25 (**Union bound inequality**, cf. [141] Lemma 1.2)**.** *For any finite or countably infinite sequence of events $E_1, E_2, \ldots$ it holds that*

$$\mathbf{Pr}\left[\bigcup_{i\geq 1} E_i\right] \leq \sum_{i\geq 1} \mathbf{Pr}[E_i].$$

The following two results relate the probability distribution of a random variable to the expected value of that variable. The Markov inequality is often too weak, but nevertheless a basic result for better bounds. The Chernoff inequality is a powerful result, whose bound decreases as an exponential function.

Lemma 2.26 (**Markov inequality**, cf. [141] Theorem 3.1)**.** *Let X be a random variable that assumes only nonnegative values. Then, for all $a > 0$,*

$$\mathbf{Pr}[X \geq a] \leq \frac{\mathbb{E}[X]}{a}.$$

Lemma 2.27 (**Chernoff inequality**, cf. [141] Theorems 4.4 and 4.5)**.** *Let $X_1, \ldots, X_n$ be independent 0-1 random variables, such that $\mathbf{Pr}[X_i = 1] = p_i$ and $\mathbf{Pr}[X_i = 0] = 1 - p_i$, for all $i \in [n]$. Let $X = \sum_{i=1}^{n} X_i$. Then, for $0 < \eta < 1$, it holds that*

$$\mathbf{Pr}[X \geq (1+\eta)\cdot\mathbb{E}[X]] \leq e^{-\eta^2\mathbb{E}[X]/3} \tag{2.19}$$

and

$$\mathbf{Pr}[X \leq (1-\eta)\cdot\mathbb{E}[X]] \leq e^{-\eta^2\mathbb{E}[X]/2}. \tag{2.20}$$

2.4 Clustering problems and coresets

Let $\mathcal{X}$ be a ground set equipped with dissimilarity measure $\mathbf{d}$. Let $n, k \in \mathbb{N}$ be positive integers. Let $P \subset \mathcal{X}$ be a finite set of points with $P = \{p_1, \ldots, p_n\}$. We want to find a set

$C \subset \mathcal{X}$ of k points, that we call centers, which minimize one of the following cost functions:

$$\mathrm{cost}_\infty(P, C) = \max_{i \in \{1,\ldots,n\}} \min_{c \in C} \mathbf{d}(p_i, c)\,, \tag{2.21}$$

$$\mathrm{cost}_1(P, C) = \sum_{i=1}^{n} \min_{c \in C} \mathbf{d}(p_i, c)\,, \tag{2.22}$$

We refer to the clustering problem as k**-center** (Equation 2.21) and k**-median** (Equation 2.22), respectively. Furthermore, we denote with $\mathrm{opt}_k^{(i)}(P)$, $i \in \{\infty, 1\}$, the cost of an optimal k-center (resp. k-median) clustering of P.

In case we have only one clustering center c, we will simply write $\mathrm{cost}_i(P, c)$ instead of $\mathrm{cost}_i(P, \{c\})$, for $i \in \{\infty, 1\}$. The k-center problem for $k = 1$ in Euclidean space is called the **smallest enclosing ball** problem.

We will occasionally in this thesis refer to the k**-means** clustering problem, related to the k-center and the k-median problems. For the same input as above, the goal is to find a set $C \subset \mathcal{X}$ of k points, that minimize

$$\mathrm{cost}_2(P, C) = \sum_{i=1}^{n} \min_{c \in C} \mathbf{d}(p_i, c)^2\,. \tag{2.23}$$

Intuitively, given clustering problem for an input set P, a strong coreset is a weighted set of points such that it approximates the clustering cost of the input by any k-tuple of the cluster centers, up to a multiplicative factor of $(1 + \varepsilon)$. We give the formal definition according to Har-Peled and Mazumdar [97].

Definition 2.28 (**Strong coreset**, cf. [97] Definition 3.1)**.** *Given $\varepsilon > 0$ and $k \in \mathbb{N}$. For a point set P in a metric space $\mathcal{X}$, a weighted set $S \subset \mathcal{X}$ is a strong (k, ε)-coreset for the k-center (resp. k-median) clustering problem, if for any set C of k points in $\mathcal{X}$, we have*

$$(1 - \varepsilon)\,\mathrm{cost}_i(P, C) \leq \mathrm{cost}_i(S, C) \leq (1 + \varepsilon)\,\mathrm{cost}_i(P, C),$$

for $i = \infty$ (resp. $i = 1$).

The requirements of Definition 2.28 are difficult to fulfill, as they require to approximate the distances to *every* potential set of centers. Sometimes it is enough to find a smaller set that will approximate the clustering cost of the whole input only for an *optimal* choice of clustering centers (up to a multiplicative factor of $(1 + \varepsilon)$). The definition of a weak coreset in the literature is not unique. We give the definition of the weak coreset for the 1-center problem in $\mathbb{R}^d$ (smallest enclosing ball), according to Bădoiu, Har-Peled, and Indyk [22],

as their definition was also used by Bădoiu and Clarkson [20, 21]. We refer to the latter result later in this thesis.

Definition 2.29 (**Weak coreset**, cf. [22])**.** *Given is $0 < \varepsilon < 1$. For a point set P in $\mathbb{R}^d$, a set $S \subset P$ is a weak ε-coreset for the 1-center clustering problem (smallest enclosing ball), if the radius of the smallest enclosing ball of S is at least $1/(1+\varepsilon)$ times the radius of the smallest enclosing ball of P. The center c^* of the smallest enclosing ball of S satisfies that*

$$\operatorname{opt}_1^{(\infty)}(P) \leq \operatorname{cost}_\infty(P, c^*) \leq (1+\varepsilon)\operatorname{opt}_1^{(\infty)}(P).$$

2.4.1 Related work on clustering problems

Hardness results

The k-**center** problem in $\mathbb{R}^d$ under the ℓ_∞-norm, as well as under the ℓ_1- and the ℓ_2-norm, is **NP**-hard for $d \geq 2$, as shown by Feder and Greene [80]. They have also shown that even approximating the optimal cost within a factor smaller than 2 (under ℓ_∞ and ℓ_1), and a factor smaller than 1.822 under the ℓ_2-norm is **NP**-hard. Some of the results of Feder and Greene [80] can be obtained from an earlier result by Megiddo and Supowit [138]. Note that Megiddo [137] showed that the 1-center problem in $\mathbb{R}^2$ can be solved in linear time. The k-center problem in the general metric spaces is also **NP**-hard to approximate within a factor of 2, as shown by Hochbaum and Shmoys [101].

The k-**median** problem in $\mathbb{R}^d$ under the ℓ_1- and the ℓ_2-norm was shown to be **NP**-hard for $d \geq 2$ by Megiddo and Supowit [138]. The isometric embedding of $(\mathbb{R}^2, \ell_1)$ into $(\mathbb{R}^2, \ell_\infty)$, given in Lemma 2.8 for $d = 2$, implies that the k-median problem is also **NP**-hard under ℓ_∞ for $d = 2$ (and therefore for $d \geq 2$).

For the k-median problem its *discrete* version is often observed, i.e. if the centers are to be chosen from the set of the input points. The discrete version of the k-median problem in $\mathbb{R}^d$ was shown by Papadimitriou [150] to be **NP**-hard as well. This implies that both versions of the k-median problem are **NP**-hard in the general metric settings as well.

Even to approximate the k-median solution by a polynomial time algorithm in general metric spaces is not possible by a factor better than $1 + 2/e \approx 1.736$ unless **NP** $\subseteq$ DTIME $\left[n^{O(\log \log n)}\right]$. This was shown by Jain, Mahdian and Saberi [115].

k-clustering in general metric spaces

There is a number of approximation algorithms for both k-center and k-median problems. For the k-**center** problem in general metric spaces, a simple greedy 2-approximation algorithm in time linear in the input was given independently by Gonzalez [89] and by

Hochbaum and Shmoys [101], which is also optimal. We utilize this algorithm in Section 4.3, and state it as Theorem 4.4.

For the k-**median** problem we state the approximation results for the discrete version of the problem, as that one is mostly researched. Because of the triangle inequality, every α-approximation for the discrete version is a (2α)-approximation for the general version of the k-median problem.

The first constant-factor approximation algorithm for the (discrete) k-median problem in general metric spaces was given by Charikar, Guha, Tardos and Shmoys [53]. They obtained the approximation factor of $6\frac{2}{3}$ in time polynomial in n and k. Mettu and Plaxton [140] gave a lower bound on the running time of any (deterministic or randomized) constant-factor approximation algorithm of $\Omega(nk)$, and an $O(1)$-approximation algorithm. Chen [54] presented a $(10+\varepsilon)$-approximation algorithm with time linear in n, and polynomial in k. He used strong coresets of size $O\left(dk^2\varepsilon^{-2}\log n\log(k/\varepsilon)\right)$. We use the result of Chen [54] in Section 4.3, and state it as Theorem 4.5.

The approximation factor after the result of Chen, was further improved by two recent results, but with a running time no longer linear in n. Li and Svensson [130] gave a $(1+\sqrt{3}+\varepsilon)$-approximation in time $O\left(n^{(1/\varepsilon)^2}\right)$. Byrka *et al.* [50] improved the result of Li and Svensson from $2.732+\varepsilon$ to a $(2.675+\varepsilon)$-approximation algorithm, with the running time $O\left(n^{(1/\varepsilon)\log(1/\varepsilon)}\right)$.

k-clustering in Euclidean spaces

In the Euclidean space $\mathbb{R}^d$, one is usually interested in a $(1+\varepsilon)$-approximation algorithm for both k-center and k-median problems. There exists a series of such algorithms, and we address only those with the running time at most linear in n. Many of the algorithms we address are based on coresets. For a survey of the coreset methods we confer to the work of Munteanu and Schwiegelshohn [144].

For the k-**center** problem, the first $(1+\varepsilon)$-approximation algorithm was given by Bădoiu, Har-Peled and Indyk [22]. Their algorithm constructs a (weak) coreset of size $O\left(1/\varepsilon^2\right)$ for the 1-center problem, and then uses this coreset to find a candidate set for the solution of the k-center problem. The running time of their algorithm is $2^{O\left((k\log k)/\varepsilon^2\right)}\cdot dn$. For the 1-center problem, this yields an $O\left(dn/\varepsilon^2+(1/\varepsilon)^{O(1)}\right)$-time algorithm.

Bădoiu and Clarkson [20] improved the result of Bădoiu, Har-Peled and Indyk [22], by reducing the coreset size for the 1-center problem to $\lceil 2/\varepsilon\rceil$. Bădoiu and Clarkson [21] showed that $\lceil 1/\varepsilon\rceil$ is the optimal (weak) coreset size for the 1-center problem. The coresets of Bădoiu and Clarkson [20, 21] for the 1-center problem imply the small coresets for the k-center problem, and yield the $(1+\varepsilon)$-approximation algorithm to the k-center

problem in time $2^{O((k \log k)/\varepsilon)} \cdot dn$. For the 1-center problem, the running time is reduced to $O\left(dn/\varepsilon + (1/\varepsilon)^5\right)$.

Based on the $(1+\varepsilon)$-approximation for the discrete k**-median** by Kolliopoulos and Rao [120], Har-Peled and Mazumdar [97] gave a $(1+\varepsilon)$-approximation algorithm without the assumption on the centers' choice with running time $O\left(n + \text{poly}(k, \log n) \cdot \exp(\text{poly}(1/\varepsilon))\right)$. The algorithm of Har-Peled and Mazumdar uses (strong) coresets of size $O\left((k \log n)/\varepsilon^d\right)$. Har-Peled and Kushal [96] constructed the coresets whose size does not depend on n: $O\left(k^2/\varepsilon^d\right)$. This improves the running time of the $(1+\varepsilon)$-approximation algorithm to $O\left(n + \text{poly}(k, \log n, 1/\varepsilon)\right)$.

Another line of research is built upon the $(1+\varepsilon)$-approximation algorithm for the k-median problem by Kumar, Sabharwal and Sen [126] in time $O\left(nd \cdot 2^{(k/\varepsilon)^{O(1)}}\right)$, based on the *random sampling*. Chen [54] improved the result of Kumar, Sabharwal and Sen [126] using coresets of size $O\left(dk^2 \log n \log(k/\varepsilon)/\varepsilon^2\right)$, with a total running time of $O\left(nd + 2^{(k/\varepsilon)^{O(1)}} \cdot d^2 \log^{k+2} n\right)$. Feldman and Langberg [81] improved this result further using coresets of size $O\left((dk \log k)/\varepsilon^2\right)$, and reducing the running time to $O\left(nd + 2^{\text{poly}(k,1/\varepsilon)}\right)$. Only recently, the strong coresets for the Euclidean k-median problem of size independent of the dimension: $O\left((k^2 \log k)/\varepsilon^4\right)$, were presented by Sohler and Woodruff [163]. The coresets of Sohler and Woodruff can be computed in time $\tilde{O}\left((n+d)\,\text{poly}(k/\varepsilon) + \exp(\text{poly}(k/\varepsilon))\right)$.

Kumar, Sabharwal and Sen [126] showed that a small uniform sample of a constant number of input points, independent of n: $O\left((1/\varepsilon)^{O(1)}\right)$, is sufficient to construct a candidate set of size $O\left(2^{(1/\varepsilon)^{O(1)}}\right)$, that contains a $(1+\varepsilon)$-approximation for the 1-median. Indyk and Thorup [110, 168] showed that a uniform sample of size $O\left((1/\varepsilon^2) \cdot \log n\right)$ is sufficient to approximate the discrete metric 1-median on n points within a factor of $(1+\varepsilon)$. Ackermann, Blömer and Sohler [3] showed how this argument can be adapted to the metric spaces with finite doubling dimension, which include the continuous Euclidean space ℓ_2^d. We adapt these ideas to find a $(1+\varepsilon)$-approximation in Section 6.2.

Ackermann, Blömer and Sohler [3] showed that a $(1+\varepsilon)$-approximation to the k-median problem in general metric spaces can be efficiently found, if a $(1+\varepsilon)$-approximation to the 1-median can be found by taking a random sample of constant size, and exactly solving the 1-median problem on the sample. This result holds not only for the metric spaces with finite doubling dimension (e.g. ℓ_2^d), but also for the (not necessarily metric) spaces, whose dissimilarity measure satisfies the *sampling property* (cf. Theorem 4.14). We utilize this result for our clustering algorithm in Section 4.5.

Subgradient descent method and the Euclidean 1-clustering problems

The stochastic subgradient descent is a popular and often only implicitly used technique in the coreset literature, and a method for solving the 1-center (smallest enclosing ball) and the 1-median problem. It is derived from convex optimization, and we give a brief overview of this method in Section 2.2.

One of the first coreset constructions using stochastic subgradient descent was given in the uniform sampling algorithm of Bădoiu, Har-Peled and Indyk [22] for 1-median. In each iteration a single point is sampled uniformly. Moving the current center towards that point, for a carefully chosen step size, improves the solution with high probability. Each step can be seen as taking a descent in a uniformly random direction from the subgradient which roughly equals the sum of directions to all points. The result is a set of candidate solutions of size $O(2^{(1/\varepsilon)^{O(1)}} \cdot \log n)$ to $(1+\varepsilon)$-approximate the 1-median problem for the input of n points in $\mathbb{R}^d$.

The coreset construction of Bădoiu and Clarkson [20] for the 1-center problem on an input set $P \subset \mathbb{R}^d$ collects the candidate centers during a subgradient descent. Starting from an initial center, the current point of their algorithm is iteratively moved a little towards the input point that is farthest away. In their algorithm, the next point included in the (weak) coreset is the one maximizing the distance to the current best center. Note that a suitable subgradient at the current center points exactly into the opposite direction. More precisely, if $q \in P$ is a point that is farthest away from the current center c, then $(c-q)/\|c-q\| \in \partial \max_{p \in P} \|c-p\|$. The algorithm of [20] can thus be interpreted as a subgradient descent minimizing $\max_{p \in P} \|c-p\|$. The authors of [20] also gave a more explicit application of the subgradient descent to the problem with a slightly larger number of iterations.

First application of the subgradient descent to solve the 1-median was done by Weiszfeld [174, 175] in 1937. He gave an algorithm for the 1-median problem with $O(1/\varepsilon)$ subgradient iterations, and obtained an additive $O(\varepsilon)$-error. For the review of the results based on the Weiszfeld method we confer to the work of Beck and Sabach [25], who emphasized the importance of bounding the Lipschitz constant of the cost function for the optimization process.

More recently, Cohen *et al.* [57] developed one of the fastest $(1+\varepsilon)$-approximation algorithms to date for the 1-median problem, using stochastic subgradient methods, with running time $O\left(nd + d/\varepsilon^2\right)$. We note that in the publication [57] the running time is stated without the $O(nd)$-part, which is needed to read the input. Two further crucial steps to turn the additive error into a relative error are finding a suitable starting point that achieves a constant approximation, and estimating its initial distance to the optimal

solution. We generalize these approaches for the 1-center and the 1-median problem to minimize the cost function of the set median problem, that extends both 1-center and 1-median, in Section 6.2.

Further related work on clustering of the curves is described in Subsection 4.1.3. Related work on the clustering of the probabilistic data is described in Subsection 6.1.3.

2.5 Curves and their dissimilarity measures

2.5.1 Curves

A **curve** in the Euclidean space $\mathbb{R}^d$, for $d \in \mathbb{N}$, is a continuous function $\tau : [0,1] \to \mathbb{R}^d$. The domain $[0,1]$ is chosen for the simplicity of the argumentation and presentation. It can be replaced by an arbitrary interval $[a,b]$, with $a, b \in \mathbb{R}$ and $a < b$, using an arbitrary homeomorphism that maps $[a,b]$ to $[0,1]$, and the fact that the composition of two continuous functions is a continuous function (cf. [157], Chapter 9.3). A homeomorphism is a function mapping two topological spaces, that is a bijection, continuous and its inverse function is also continuous (cf. the book by Royden [157], Chapter 11).

A **polygonal curve** is a curve such that there are the values $0 = t_1 \leq t_2 \leq \ldots \leq t_m = 1$, with $w_i = \tau(t_i)$ that we call **vertices**, and such that for all $i \in \{1, \ldots, m-1\}$ each curve segment between $\tau(t_i)$ and $\tau(t_{i+1})$ is affine, i.e.

$$\tau(t_i + x) = \left(1 - \frac{x}{t_{i+1} - t_i}\right) \cdot \tau(t_i) + \frac{x}{t_{i+1} - t_i} \cdot \tau(t_{i+1}),$$

for all $x \in [0, t_{i+1} - t_i]$. The polygonal curve segments between two consecutive vertices w_i and w_{i+1} are called **edges**, and denoted with $\overline{w_i w_{i+1}}$.

We identify the curves with their images, i.e. with $\tau([0,1]) \subseteq \mathbb{R}^d$, when it is clear from the context. In this work we consider only polygonal curves, and we will simply refer to τ as a *curve*. When defining a curve, we may write "curve $\tau : [0,1] \to \mathbb{R}^d$ with m vertices", or "curve $\tau = w_1, \ldots, w_m$". We say that such a curve τ has **complexity** m. We denote its **set of vertices** with $\mathcal{V}(\tau)$.

An alternative view to the curves is provided by the data mining community, that analyzes the signal measurements. A **time series** is a series $(w_1, t_1), \ldots, (w_m, t_m)$ of measurements $w_i \in \mathbb{R}$ of a signal taken at times $t_i \in \mathbb{R}$. We assume $0 = t_1 < t_2 < \ldots < t_m = 1$ and m is finite. A time series may be viewed as a continuous function $\tau : [0,1] \to \mathbb{R}$ by linearly interpolating $w_1, \ldots, w_m$ in order of t_i, $i = 1, \ldots m$, thus being a polygonal curve in the ambient space $\mathbb{R}$. This notation does not specify the points of time at which the measurements are taken. This is justified by the choice of the dissimilarity measures we work

with, that are formally introduced in the next subsection. In the one-dimensional ambient space $\mathbb{R}$ these dissimilarity measures depend only on the ordering of the measurements (and their values), but not on the exact points of time when the measurements are made. As our study concentrates on one-dimensional ambient space, we are not going to make a distinction between the notions of *univariate time series* and *curves in* $\mathbb{R}$. Throughout this thesis we use the notion *curve* only.

For any $t_i \leq t_j \in [0,1]$, we denote the subcurve of τ starting at $\tau(t_i)$ and ending at $\tau(t_j)$ with $\tau[t_i, t_j]$. For one-dimensional curves we define $\min(\tau[t^-, t^+]) = \min\{\tau(t) : t \in [t^-, t^+]\}$ and $\max(\tau[t^-, t^+]) = \max\{\tau(t) : t \in [t^-, t^+]\}$, to denote the minimum and maximum along a (sub)curve.

Given polygonal curve $\tau = w_1, \ldots, w_m$ in $\mathbb{R}^d$, we define the **length of the curve** $\mathcal{L}(\tau)$ as $\mathcal{L}(\tau) = \sum_{1 \leq i \leq m-1} \mathcal{L}(\overline{w_i w_{i+1}}) = \sum_{1 \leq i \leq m-1} \|w_i - w_{i+1}\|$. For two curves τ and σ it is $\mathcal{L}(\tau \cup \sigma) = \mathcal{L}(\tau) + \mathcal{L}(\sigma)$. (It is neither required that the curves τ and σ do not intersect, nor that they are concatenated.) Then for a set $S \subseteq \mathbb{R}^d$, $\mathcal{L}(\tau \cap S)$ is the length of the part of the curve τ that is contained in S. Such part of the curve τ can consist of multiple (sub)curves. Note that here we want to take in count multiple instances of the same line segment separately, and not just once, as it was the case with the Lebesgue measure for the length of the possibly intersecting intervals (cf. Subsection 2.1.5).

We need the notion of curve length to define a class of realistic input curves. The c-packed curves were introduced by Driemel, Har-Peled, and Wenk in [66], who wrote that the parameter c should measure how "unrealistic" the input curves are. The c-packedness of curves is formally defined as follows.

Definition 2.30 (**c-packed curve**, cf. [66])**.** *Given $c > 0$, a curve $\tau \in \mathbb{R}^d$ is c-packed if for any point $p \in \mathbb{R}^d$ and for any radius $r > 0$, the total length of the curve τ inside the ball $\mathbf{B}(p, r)$ in the metric space $(\mathbb{R}^d, \|\cdot\|_2)$ is at most $c \cdot r$, i.e. $\mathcal{L}(\tau \cap \mathbf{B}(p, r)) \leq c \cdot r$.*

We utilize the c-packed curves to obtain better bounds on embeddings of the curves in Chapter 5. A brief discussion on computing of the distances of c-packed curves is given at the end of Subsection 2.5.3.

2.5.2 Dissimilarity measures on curves

Given are two curves $\tau : [0,1] \to \mathbb{R}^d$ and $\sigma : [0,1] \to \mathbb{R}^d$. In some literature, the curves are observed as sets of points, and thus a dissimilarity measure defined over sets can be used. One such measure is the Hausdorff distance.

Let $\tau(t)$, $t \in [0,1]$, be a point on τ. Let the dissimilarity measure of the point $\tau(t)$ to the curve σ be defined as $d_{\overrightarrow{H}}(\tau(t), \sigma) = \min_{s \in [0,1]} \|\tau(t) - \sigma(s)\|_2$. By extending this

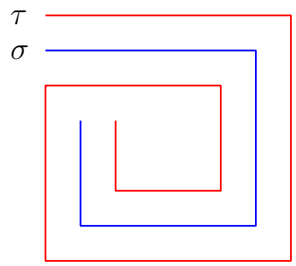


Figure 2.1: Two differently structured curves τ and σ that have small Hausdorff distance. The Hausdorff distance of these curves is small, but their Fréchet distance is much greater. A similar figure was given in [64, 66].

dissimilarity measure to the case of two curves, we have **directed Hausdorff** dissimilarity measure[4], defined as:

$$d_{\overrightarrow{H}}(\tau,\sigma) = \max_{t\in[0,1]} \min_{s\in[0,1]} \|\tau(t) - \sigma(s)\|_2.$$

Since for the two curves τ and σ it may hold that $d_{\overrightarrow{H}}(\tau,\sigma) \neq d_{\overrightarrow{H}}(\sigma,\tau)$ (see Figure 2.1 for an example), we overcome this obstacle by defining the (undirected) **Hausdorff distance** as:

$$d_H(\tau,\sigma) = \max\{d_{\overrightarrow{H}}(\tau,\sigma), d_{\overrightarrow{H}}(\sigma,\tau)\}.$$

The Hausdorff distance does not consider the inner structure of the curves and the possible curve orientation. This comes from observing the curves as sets, thus there exist curves whose structure differs a lot, but whose Hausdorff distance is small (see Figure 2.1). The distance measures of two curves that do not have such a problem, and that represent the cost of transforming one curve into the other are Fréchet and dynamic time warping distance.

Let $\mathcal{H}$ denote the set of continuous and monotonically increasing functions $f : [0,1] \to [0,1]$ with the property that $f(0) = 0$ and $f(1) = 1$. Note that the functions in $\mathcal{H}$ are bijections. The functions in $\mathcal{H}$ are called **reparametrizations**. For two given functions $\tau : [0,1] \to \mathbb{R}^d$ and $\sigma : [0,1] \to \mathbb{R}^d$, their continuous **Fréchet distance**[5] is defined as

$$d_F(\tau,\sigma) = \inf_{f\in\mathcal{H}} \max_{t\in[0,1]} \|\tau(t) - \sigma(f(t))\|_2, \tag{2.24}$$

The Fréchet distance between the two curves is defined as the Fréchet distance of their corresponding continuous functions. Note that any $f \in \mathcal{H}$ induces a bijection between the

[4]This dissimilarity measure is usually called "directed Hausdorff distance", but we defined that a distance function must be symmetric.
[5]Usually called simply "Fréchet distance".

two curves. We refer to the function f that realizes the Fréchet distance as a **matching**. We say that the matching **witnesses** the Fréchet distance between the two curves.

It may be that such a matching exists in the limit only. That is, for any $\varepsilon > 0$, there exists a $f \in \mathcal{H}$ that matches each point on τ to a point on σ within distance $d_F(\tau, \sigma) + \varepsilon$. In particular, there is a continuous function $f' : [0,1] \to [0,1]$, that is monotonically non-decreasing, with $\max_{t\in[0,1]} \|\tau(t) - \sigma(f'(t))\|_2 = d_F(\tau, \sigma)$. By a slight perturbation of f' we have a bijection $f \in \mathcal{H}$ with $\max_{t\in[0,1]} \|\tau(t) - \sigma(f(t))\|_2 = d_F(\tau, \sigma) + \varepsilon$, for arbitrarily small $\varepsilon > 0$.

Clearly by the definition of the Fréchet distance the interval $[0,1]$ can be adapted to any domain interval. The Fréchet distance is invariant under reparametrizations by an arbitrary homeomorphism.

It is well-known that the Fréchet distance is a pseudo-metric (cf. [14]), i.e. it satisfies all properties of a metric in the ambient space $\mathbb{R}^d$, except that there may be two different functions $\tau, \sigma : [0,1] \to \mathbb{R}^d$ such that $d_F(\tau, \sigma) = 0$. To resolve this, we observe all such functions to be equivalent, and consider the equivalence classes that are induced by functions of pairwise distance 0 represented by a single function.

In this thesis, particularly in Chapter 3 and Chapter 4, we work with curves in one-dimensional ambient space $\mathbb{R}$, i.e. with univariate time series. We represent the curves as an ordered list of measurements, without explicit time-stamps. We notice that only the ordering of the measurement values w_i is relevant and that under Fréchet distance two one-dimensional curves can be considered of being identical, if they have the same sequence of local minima and maxima. Therefore, we can assume that a curve is induced by its sequence of local minima and maxima and we will use the term "curve" throughout this work to describe the equivalence class of curves with pairwise Fréchet distance 0.

Therefore, we obtain a metric space (Δ, d_F), defined by the set Δ of all (equivalence classes of) one-dimensional curves and the continuous Fréchet distance. We denote with Δ_m the set of all one-dimensional curves of complexity at most m.

The continuous Fréchet distance requires for a reparametrization for the complete domain interval $[0,1]$. A related dissimilarity measure is the discrete Fréchet distance, which requires only for a mapping between vertices of the input curves. For simplicity of the notation, we adapt the domain of the functions: let the polygonal curves $\sigma : [1, m''] \to \mathbb{R}^d$ and $\tau : [1, m'] \to \mathbb{R}^d$ be given by their sequences of vertices $\sigma = v_1, \ldots, v_{m''}$ and $\tau = w_1, \ldots, w_{m'}$. We may assume that the parameters of the curves are chosen in such manner that for all $i \in [m'']$, it is $\sigma(i) = v_i$, and for all $j \in [m']$, it is $\tau(j) = w_j$.

A **traversal** T of σ and τ is a sequence of pairs of indices (i, j) of vertices $(v_i, w_j) \in \sigma \times \tau$ such that

i) the traversal T starts with $(1,1)$ and ends with (m'', m'), and
ii) the pair (i,j) of T can be followed only by one of $(i+1,j)$, $(i,j+1)$ or $(i+1,j+1)$.

We notice that every traversal is monotone. If $\mathcal{T}$ is the set of all traversals T of σ and τ, then the **discrete Fréchet distance** between σ and τ is defined as

$$d_{dF}(\sigma,\tau) = \min_{T\in\mathcal{T}} \max_{(i,j)\in T} \|v_i - w_j\|_2. \tag{2.25}$$

The discrete Fréchet distance is a metric on the set of polygonal curves in $\mathbb{R}^d$ (cf. [75] Proposition 1). In particular, the difference to the continuous case is that two curves can be at the discrete Fréchet distance 0 if and only if they are equal.

A related dissimilarity measure between the two curves to the discrete Fréchet distance is **dynamic time warping** distance. It considers the *sum* of the used distances in the traversal (instead of the maximum distance). Formally, for the two curves σ and τ from the ambient space $\mathbb{R}^d$, as given for the discrete Fréchet distance, we define:

$$d_{DTW}(\sigma,\tau) = \min_{T\in\mathcal{T}} \sum_{(i,j)\in T} \|v_i - w_j\|_2. \tag{2.26}$$

The dynamic time warping distance is not a metric, as it does not satisfy the identity of indiscernible elements and the triangle inequality.

2.5.3 Computing the Fréchet distance

To compute the continuous Fréchet distance, the algorithm of Alt and Godau [14] is commonly used. We give a brief overview of the algorithm next.

For two curves $\tau : [0,1] \to \mathbb{R}^d$ and $\sigma : [0,1] \to \mathbb{R}^d$ their **parametric space** is $[0,1]\times[0,1]$. Then for a given parameter $\Theta \geq 0$, the **Θ-free-space** of τ and σ is defined as

$$F_\Theta(\tau,\sigma) = \{(t,s) \in [0,1]\times[0,1] \colon \|\tau(t) - \sigma(s)\|_2 \leq \Theta\}. \tag{2.27}$$

The parametric space can be divided into a grid called the free-space diagram. The vertical lines of this grid correspond to the vertices of τ, and the horizontal lines correspond to the vertices of σ. Every free-space cell in this grid corresponds to a pair of edges, one from the polygonal curve τ and another from the polygonal curve σ. For a fixed $\Theta > 0$ the free-space has properties utilized by the algorithm of Alt and Godau, and which we state as Lemma 2.31.

Lemma 2.31 (cf. [14] Lemma 3 and 4)**.** *Given are two polygonal curves τ and σ and a parameter $\Theta > 0$. For each pair of edges from τ and σ, the free-space in the corresponding*

free-space cell is the intersection of the cell rectangle with an ellipse (possibly degenerated to a line), and is thus convex. It is $d_F(\tau,\sigma) \leq \Theta$ if and only if there exists a curve within corresponding free-space diagram $F_\Theta(\tau,\sigma)$ from $(0,0)$ to $(1,1)$ which is monotonically non-decreasing in both coordinates.

Using Lemma 2.31 for two curves τ and σ of complexities m' and m'' respectively we can decide (using dynamic programming) in time $O(m'm'')$ if $d_F(\tau,\sigma) \leq \Theta$. The value of $d_F(\tau,\sigma)$ can be computed using a parametric search technique over the critical values, for which the structural changes happen to the free-space diagram, and thus obtaining the following lemma.

Theorem 2.32 (**Alt and Godau algorithm** cf. [14] Theorem 6)**.** *For given polygonal curves τ and σ, with complexities m' and m'', respectively, there is an algorithm that computes the (continuous) Fréchet distance $d_F(\tau,\sigma)$ in time $O(m'm'' \log(m'm''))$.*

To compute the discrete Fréchet distance and the dynamic time warping distance, it suffices to use dynamic programming. We state the result for the discrete Fréchet distance, provided originally by Eiter and Mannila [75].

Theorem 2.33 (**Eiter and Mannila algorithm**, cf. [75] Theorem 7)**.** *For given polygonal curves τ and σ, with complexities m' and m'', respectively, there is an algorithm that computes the discrete Fréchet distance $d_{dF}(\tau,\sigma)$ in time $O(m'm'')$.*

Theorem 2.33 holds for the dynamic time warping distance as well. The DTW distance can be computed in time $O(m'm'')$ using dynamic programming (cf. [88]).

These algorithms, although a quarter of century old, remain almost state-of-the-art and are frequently algorithms of choice. Let us assume that both input curves have complexity m. For the continuous Fréchet distance, there is an algorithm of Har-Peled and Raichel [98] that has the same asymptotical running time as the algorithm of Alt and Godau (Theorem 2.32). The advantage of this algorithm over the algorithm of Alt and Godau is that it avoids using the parametric search technique, that can generate large constants during execution. The best known algorithm to compute the continuous Fréchet distance was given by Buchin, Buchin, Meulemans and Mulzer [40], which has expected running time $O\left(m^2\sqrt{\log m}\,(\log\log m)^{3/2}\right)$ using real RAM model, and expected running time $O\left(m^2\,(\log\log m)^2\right)$ using word RAM model.[6]

For the discrete Fréchet distance, the best algorithm was given by Agarwal, Ben Avraham, Kaplan and Sharir [5], and has the running time $O\left(m^2 \log\log m/\log m\right)$ using word RAM model. The dynamic time warping distance can be deterministically computed in

[6]For distinction between two RAM models we refer to [40].

$O\left(m^2 \log\log\log m / \log\log m\right)$ time using real RAM model. This was shown by Gold and Sharir [88].

The only known lower bound for both the discrete and the continuous Fréchet distance was given by Buchin *et al.* [38]. They showed that the time $\Omega(m \log m)$ is needed for the problem of deciding whether the (continuous or discrete) Fréchet distance of two curves in the space $\mathbb{R}^2$ is at most a given value. Alt [13] conjectured that the decision problem of continuous Fréchet distance is 3SUM-hard. Since Alt's conjecture, both affirmative (by Grønlund and Pettie [90]) and negative (by Buchin, Buchin, Meulemans and Mulzer [40]) arguments were presented. However, a definite answer is not yet known.

More fruitful was research on conditional lower bounds for the Fréchet distance problem based on the Strong Exponential Time Hypothesis (SETH), conjectured by Impagliazzo, Paturi and Zane (cf. [108, 109]), that we state next (as it was presented by Bringmann [31]).[7]

Hypothesis 2.34 (**SETH**, cf. [108, 109])**.** *There is no $\eta > 0$, such that there is an algorithm for all k, with running time $O\left((2-\eta)^N\right)$, that answers if a formula in conjunctive normal form with N variables and whose each clause is limited to at most k literals, is satisfiable.*

Bringmann [31] showed that, unless SETH fails, there is no $O\left(m^{2-\eta}\right)$ algorithm to compute the (continuous or discrete) Fréchet distance for any $\eta > 0$, in the ambient space $\mathbb{R}^d$, $d \geq 2$. This result was extended by Bringmann and Mulzer [37] for the discrete Fréchet distance and for $d = 1$. For the continuous Fréchet distance in one-dimensional ambient space, no lower bounds are known. Buchin *et al.* [43] stated that the computing of the continuous Fréchet distance problem has a special structure in one-dimensional ambient space. It was independently shown by Bringmann and Künnemann [33] and by Abboud, Bačkurs and Williams [2], that there is no algorithm for the dynamic time warping distance in time $O\left(n^{2-\eta}\right)$ for any $\eta > 0$, unless SETH fails.

If an approximation scheme for these distance measures is of interest, the following results have been reported. Bringmann [31] showed that the (continuous and discrete) Fréchet distance cannot be approximated better than the factor 1.001 by an algorithm with running time $O\left(m^{2-\eta}\right)$, for any $\eta > 0$, unless SETH fails. This was improved for the discrete Fréchet distance by Bringmann and Mulzer [37], who showed that any 1.399-approximation algorithm in time $O\left(m^{2-\eta}\right)$ in one-dimensional ambient space violates SETH.

[7] Besides SETH, in [108, 109] Exponential Time Hypothesis (ETH) was introduced. ETH asserts that 3-SAT problem has no $2^{o(N)}$-time algorithm. SETH is stronger than ETH. Bringmann [31] stated that ETH is not suited for proving polynomial time bounds, since it does not specify the exponent.

However, there are some positive approximation results. A simple greedy algorithm by Bringmann and Mulzer [37] provides a $2^{\Theta(m)}$-approximation of the discrete Fréchet distance in time $O(m)$. This algorithm extends to the continuous Fréchet distance case, with the same approximation guarantee and the running time.

Bringmann and Mulzer [37] gave an α-approximation algorithm for the discrete Fréchet distance as well. It has a running time $O\left(m \log m + m^2/\alpha\right)$, for any $1 \leq \alpha \leq m$. This implies that if $\alpha = m/\log m$, then the α-approximation is obtained in time $O(m \log m)$. An (m^η)-approximation is obtained in time $O\left(m^{2-\eta}\right)$, for any $0 < \eta < 1$. Therefore, a much better approximation compared to the greedy algorithm is obtained, at the cost of the running time.

If additional assumptions on the input curves are made, then faster algorithms are possible. Recently, it was reported by Kuszmaul [127], that $d_{DTW}(\tau, \sigma)$ between two curves τ and σ of complexity m in general metric ambient space $\mathcal{X}$ can be computed in time $O(m \cdot d_{DTW}(\tau, \sigma))$. The result of Kuszmaul has a caveat, that the smallest non-zero distance between two points in $\mathcal{X}$ is normalized to 1. This is not a problem if the input are two strings, but for the curves in $\mathbb{R}^d$ this assumption is not negligible.

For the realistic class of curves we consider in this thesis – if the curves are c-packed for some constant $c > 0$, then efficient $(1+\varepsilon)$-approximation algorithms for all three of these distance measures exist. A $(1+\varepsilon)$-approximation to the continuous Fréchet distance can be computed in $O(cm/\varepsilon + cm \log m)$ time, as shown by Driemel, Har-Peled, and Wenk [66], by exploring the complexity of the free-space diagram. Their result was improved by Bringmann and Künnemann [34], who gave an $O\left((cm/\sqrt{\varepsilon}) \cdot \log^2(1/\varepsilon) + cm \log m\right)$-time $(1+\varepsilon)$-approximation algorithm. The result of Bringmann and Künnemann [34] is actually optimal in high dimensions, unless SETH fails. This follows from the result by Bringmann [31], who showed that for $d \geq 5$ there are no $(1+\varepsilon)$-approximation algorithms for the continuous Fréchet distance between two c-packed curves in time $O\left((cm/\sqrt{\varepsilon})^{1-\eta}\right)$ for any $\eta > 0$, unless SETH fails.

The algorithm of Bringmann and Künnemann [34] extends for the discrete Fréchet distance. The running time of their $(1+\varepsilon)$-approximation algorithm for the c-packed curves in the discrete case is $O((cm/\sqrt{\varepsilon}) \cdot \log(1/\varepsilon) + cm \log m)$. Analogously to the continuous case, this is near-optimal, except in the low dimensions.

A $(1+\varepsilon)$-approximation for the dynamic time warping distance of two c-packed curves was given by Agarwal, Fox, Pan and Ying [6]. Their algorithm runs in $O((cm/\varepsilon) \cdot \log m)$ time. It is not known if a (conditional) lower bound for this problem exists.

2.5.4 Notes on the concatenation of curves

As we are going to construct new matchings of curves to maintain the Fréchet distance, we formally define the notion of concatenation of two curves in $\mathbb{R}^d$.

Definition 2.35 (Concatenation). *Let two curves $\tau_1 : [a_1, b_1] \to \mathbb{R}^d$, $0 \leq a_1 \leq b_1 \leq 1$, and $\tau_2 : [a_2, b_2] \to \mathbb{R}^d$, $0 \leq a_2 \leq b_2 \leq 1$ be given, such that $\tau_1(b_1) = \tau_2(a_2)$. The concatenation of τ_1 and τ_2 is a curve τ defined as $\tau = \tau_1 \oplus \tau_2 : [0, 1] \to \mathbb{R}^d$, such that*

$$\tau(t) = (\tau_1 \oplus \tau_2)(t) = \begin{cases} \tau_1\left(a_1 + (b_1 - a_1 + b_2 - a_2) \cdot t\right) & \text{if } t \leq \frac{b_1 - a_1}{b_1 - a_1 + b_2 - a_2} \\ \tau_2\left(b_2 - (b_1 - a_1 + b_2 - a_2) \cdot (1 - t)\right) & \text{if } t > \frac{b_1 - a_1}{b_1 - a_1 + b_2 - a_2}. \end{cases}$$

We are going to use the following simple lemmas in Chapter 3, often without explicitly referring to them. We present their proofs for completeness.

Lemma 2.36. *Let two curves $\tau : [0, 1] \to \mathbb{R}^d$ and $\pi : [0, 1] \to \mathbb{R}^d$ be the concatenations of two subcurves: $\tau = \tau_1 \oplus \tau_2$ and $\pi = \pi_1 \oplus \pi_2$, then it holds that*

$$d_F(\tau, \pi) \leq \max\{d_F(\tau_1, \pi_1), d_F(\tau_2, \pi_2)\}.$$

Proof. Let the parameter $\hat{t} \in [0, 1]$ be such that $\tau(t) = \tau_1(t)$ for all $0 \leq t \leq \hat{t}$, and $\tau(t) = \tau_2(t)$ otherwise, as in Definition 2.35. Let the matchings $f_1, f_2 : [0, 1] \to \mathbb{R}^d$ witness $d_F(\tau_1, \pi_1)$ and $d_F(\tau_2, \pi_2)$, respectively. Let the mapping $f : [0, 1] \to [0, 1]$ be defined as

$$f(t) = \begin{cases} f_1(t) & \text{if } t \leq \hat{t} \\ f_2(t) & \text{if } t > \hat{t}. \end{cases}$$

This mapping is a continuous and monotonically increasing mapping from τ to π. It witnesses that the Fréchet distance between τ and π is at most $\max\{d_F(\tau_1, \pi_1), d_F(\tau_2, \pi_2)\}$, as claimed. □

Lemma 2.37. *Given two edges $\overline{a_1 a_2}$ and $\overline{b_1 b_2}$ with $a_1, a_2, b_1, b_2 \in \mathbb{R}^d$, it holds that*

$$d_F\left(\overline{a_1 a_2}, \overline{b_1 b_2}\right) = \max\{|a_1 - b_1|, |a_2 - b_2|\}.$$

Proof. The Fréchet distance between the two edges is at least $\vartheta = \max\{|a_1 - b_1|, |a_2 - b_2|\}$, as the endpoints must be pairwise matched. But the Fréchet distance has the value ϑ at most. Namely, by Lemma 2.31, any free-space diagram of the edges $\overline{a_1 a_2}$ and $\overline{b_1 b_2}$ is the intersection of rectangle (the only cell) and an ellipse, and thus is a convex set. Since both

pairs $(0,0)$ and $(1,1)$ are in the ϑ-free-space, by convexity, the straight line connecting the endpoints $(0,0)$ and $(1,1)$ is in the ϑ-free-space. This line is monotonically non-decreasing in both coordinates, and thus $d_F\big(\overline{a_1a_2}, \overline{b_1b_2}\big) \leq \vartheta$. □

Lemma 2.36 holds for the discrete Fréchet distance as well, by the same argument over traversals instead of the matchings. Lemma 2.37 holds for the discrete Fréchet distance by definition.

3 Curve simplification under the Fréchet distance

3.1 Introduction

The curve simplification problem has been studied under different names, for multidimensional curves and under various error measures, in many scientific domains, such as cartography [63, 154], computational geometry [86], data mining [118], pattern recognition [153], and structural biology [78]. In some of these areas, a large body of work is commited to a search for a fast heuristic or a solution that works well for an average case. However, a mathematically provable worst-case analysis is always welcome.

Given a curve τ, a simplification σ of τ is a curve which has a lower complexity than the original curve, and which is similar to the original curve. The dissimilarity between τ and σ is measured by a distance measure $\mathbf{d}$, where the *error* $\mathbf{d}(\tau, \sigma)$ needs to be small. There are many variants of this problem. Some of them require vertex-constraint simplifications, i.e. that the vertices of σ come from the set of the vertices of τ, and additionally, that they respect the same order in τ and σ, and/or that the endpoints are kept. If there are no conditions on the vertices of σ, such simplifications are called weak.

In this chapter we introduce a special type of the curve simplification – the signatures. A signature is a vertex-constrained simplification of the given polygonal curve in the one-dimensional Euclidean ambient space. The error of a signature is measured by the continuous Fréchet distance. The exact definition of the signature (cf. Definition 3.3) is somewhat cumbersome. However, as it will turn out, such a definition provides exact properties to prove the technical lemmas, and consequently, that enable us to construct efficient clustering algorithms for the curves in one-dimensional ambient space (cf. Chapter 4). The signatures are envisioned to be a tool, that is not bounded only to the usage in clustering problems, and as such of independent interest. Unfortunately, they are bounded to work only in one-dimensional space, as their properties heavily depend on the features of the space $\mathbb{R}$.

In this chapter we use the following non-standard notation:

- Let $\langle\langle a, b \rangle\rangle = [\min(a, b), \max(a, b)]$, for any $a, b \in \mathbb{R}$.

- Let $[h]_\delta = [h - \delta, h + \delta]$, for any $h \in \mathbb{R}$ and $\delta > 0$.
- For a given curve τ, and two parameters t' and t'', with $0 \leq t' \leq t'' \leq 1$, we denote $\max(\tau[t', t'']) = \max\{\tau(t) \colon t \in [t', t'']\}$, and $\min(\tau[t', t'']) = \min\{\tau(t) \colon t \in [t', t'']\}$.

3.1.1 Definition of the signatures

In a general setting, the curve simplification problem is not bounded by any requirements. The simplification is a bicriteria optimization problem. The quality of the solution is measured by two parameters: error and size. Minimizing each of them separately defines an optimization problem. We state these problems as Definition 3.1 and Definition 3.2, as it is done in the standard literature.

Definition 3.1 (**Minimum-error ℓ-simplification**). *A curve π is a minimum-error ℓ-simplification of τ if the complexity of π is at most ℓ and for any curve π' of at most ℓ vertices, it holds that $d_F(\pi', \tau) \geq d_F(\pi, \tau)$.*

Definition 3.2 (**Minimum-size ε-simplification**). *A curve π is a minimum-size ε-simplification of τ if $d_F(\pi, \tau) \leq \varepsilon$ and for any curve π' such that $d_F(\pi', \tau) \leq \varepsilon$, it holds that the complexity of π' is at least as much as the complexity of π.*

The signatures are none of these two, but provide a good approximation of the optimal solutions for both the minimum-error and the minimum-size simplification problems. Our definition aligns with the work by Pratt and Fink [153] on computing important minima and maxima in the context of time series compression. Intuitively, the signatures provide us with the "shape" of a curve (in their work Pratt and Fink call the curve time series) at multiple scales. A signature has a parameter $\delta > 0$. This parameter intuitively describes the minimum-edge-length of the simplified curve (more than 2δ for the edges not including the endpoints), as well as the maximal direction-preserving discrepancy (on how far can two vertices of the input curve be, to be safe to ignore them in the simplified curve). The formal definition of a δ-signature is provided by Definition 3.3.

Definition 3.3 (**δ-signature**). *Given are a curve $\tau : [0, 1] \to \mathbb{R}$ and a parameter $\delta > 0$. The δ-signature of the curve τ is a curve $\sigma : [0, 1] \to \mathbb{R}$ defined by a series of values $0 = t_1 < \cdots < t_\ell = 1$ as the linear interpolation of $\tau(t_i)$ in the order of the index i, and such that for $1 \leq i \leq \ell - 1$ the following conditions hold:*

(i) (non-degeneracy property) if $i \in [2, \ell - 1]$ then $\tau(t_i) \notin \langle\!\langle \tau(t_{i-1}), \tau(t_{i+1}) \rangle\!\rangle$;

(ii) (direction-preserving property)
if $\tau(t_i) < \tau(t_{i+1})$ for $t < t' \in [t_i, t_{i+1}]$: $\tau(t) - \tau(t') \leq 2\delta$, and
if $\tau(t_i) > \tau(t_{i+1})$ for $t < t' \in [t_i, t_{i+1}]$: $\tau(t') - \tau(t) \leq 2\delta$;

(iii) (minimum-edge-length property)
if $i \in [2, \ell-2]$ *then* $|\tau(t_{i+1}) - \tau(t_i)| > 2\delta$, *and*
if $i \in \{1, \ell-1\}$ *then* $|\tau(t_{i+1}) - \tau(t_i)| > \delta$;
(iv) (range property) for $t \in [t_i, t_{i+1}]$:
if $i \in [2, \ell-2]$ *then* $\tau(t) \in \langle\!\langle \tau(t_i), \tau(t_{i+1}) \rangle\!\rangle$, *and*
if $i = 1$ *and* $\ell > 2$ *then* $\tau(t) \in \langle\!\langle \tau(t_i), \tau(t_{i+1}) \rangle\!\rangle \cup \langle\!\langle \tau(t_i) - \delta, \tau(t_i) + \delta \rangle\!\rangle$, *and*
if $i = \ell - 1$ *and* $\ell > 2$ *then* $\tau(t) \in \langle\!\langle \tau(t_i), \tau(t_{i+1}) \rangle\!\rangle \cup \langle\!\langle \tau(t_{i+1}) - \delta, \tau(t_{i+1}) + \delta \rangle\!\rangle$, *and*
if $i = 1$ *and* $\ell = 2$ *then* $\tau(t) \in \langle\!\langle \tau(t_1), \tau(t_2) \rangle\!\rangle \cup \langle\!\langle \tau(t_1) - \delta, \tau(t_1) + \delta \rangle\!\rangle \cup \langle\!\langle \tau(t_2) - \delta, \tau(t_2) + \delta \rangle\!\rangle$.

It follows from the properties (i) and (iv) of Definition 3.3 that the parameters t_i for $i \in [\ell]$ specify vertices of τ. Furthermore, it follows that the vertex $\tau(t_i)$ is either a minimum or maximum on $\tau[t_{i-1}, t_{i+1}]$ for $2 \leq i \leq \ell - 1$.

For a signature σ we will simply write *signature* $\sigma : [0,1] \to \mathbb{R}$ *with* ℓ *vertices* or *signature* $\sigma = v_1, \ldots, v_\ell$, instead of *signature* $\sigma : [0,1] \to \mathbb{R}$, *with vertices* $v_1 = \sigma(s_1), \ldots, v_\ell = \sigma(s_\ell)$, *where* $0 = s_1 < \ldots < s_\ell = 1$. We assume that the parametrization of σ is chosen such that $\sigma(s_j) = \tau(s_j)$, for any $j \in [\ell]$. An example of a δ-signature is provided in Figure 3.1.

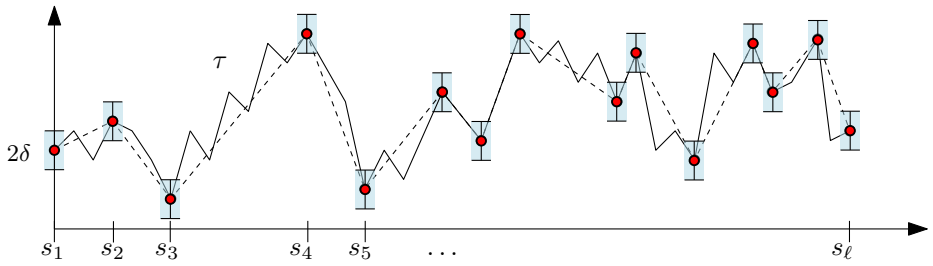


Figure 3.1: Example of a δ-signature σ of a curve τ. The vertices of σ are colored red. The ranges of width 2δ centered at the signature vertices are colored light blue. Note that the ranges of two consecutive signature vertices do not intersect, except possibly for the start- and endpoint.

In this chapter, and subsequently in Chapter 4, we make the following **general position assumption** on the input curves: for every input curve τ we assume that no two vertices τ have the same coordinates and any two differences between coordinates of two vertices of τ are different. This assumption can easily be achieved by symbolic perturbation. This is a well-known technique to cope with degenerate cases in geometric algorithms given by Edelsbrunner and Mücke [73]. Furthermore, we assume that τ has no edges of length zero and its vertices are an alternating sequence of minima and maxima, i.e. no vertex lies in the linear interpolation of its two neighboring vertices.[8]

[8] See page 4: the continuous Fréchet distance between two curves in $\mathbb{R}^1$ is completely determined by the sequence of local extrema of the two curves.

3.1.2 Results in this chapter

In Section 3.2 we show that the curves similar to the input curve $\tau : [0,1] \to \mathbb{R}$ need to be similar to its δ-signature σ, regarding both the continuous Fréchet distance to τ, and the complexity of such a curve. The main result of Section 3.2 is Theorem 3.8. This theorem claims that for a given curve τ and for any curve $\pi : [0,1] \to \mathbb{R}$, such that $d_F(\tau, \pi) \leq \delta$, we can omit the vertices of π, which are far from the vertices of the δ-signature σ of τ, while keeping the Fréchet distance of the such obtained curve to the curve τ at most δ. Theorem 3.8 follows from Lemma 3.7. As the proof of Lemma 3.7 is long and consists of a comprehensive case analysis, it is separated into Section 3.3, for the sake of readability.

The signatures always exist. In Section 3.4 we give an algorithm to compute signatures efficiently. For a given parameter δ and a given curve τ of complexity m, it is possible to construct a δ-signature of τ in time $O(m)$ (cf. Theorem 3.40). Our signatures have a unique hierarchical structure. This structure allows the construction of a data structure in time $O(m \log m)$ and that uses space $O(m)$, such that given a parameter $\ell \in \mathbb{N}$ it is possible to compute a signature of complexity ℓ in time $O(\ell \log \ell)$ (cf. Theorem 3.39). We show that our signatures are a constant-factor approximation solution to the minimum-error ℓ-simplification problem in Lemma 3.41.

3.1.3 Related work

Historically, the first minimal-size curve simplification algorithm was a heuristic algorithm independently suggested in the 1970's by Ramer [154] and Douglas and Peucker [63] and it remains popular in the area of geographic information science until today (e.g. for marine traffic pattern recognition [178]). It uses the Hausdorff error measure and has running time $O\left(m^2\right)$ (where m denotes the complexity of the input curve), but does not offer a bound to the size of the simplified curve. Recently, worst-case and average-case lower bounds on the number of vertices obtained by this algorithm were proven by Daskalakis, Diakonikolas and Yannakakis [60]. Imai and Iri [107] solved both the minimum-error and minimum-size simplification problem under the Hausdorff distance by modeling it as a shortest path problem in directed acyclic graphs. For both algorithms: of Douglas and Peucker, and of Imai and Iri, it was shown by van Kreveld, Löffler and Wiratma [172] that they may produce far-from-optimal results.

A concept similar to the simplification, called *segmentation*, of time series has been extensively studied in the area of data mining [27, 100, 167]. The standard approach for computing exact segmentations is to use dynamic programming which yields a running time of $O\left(m^2\right)$.

Curve simplification using the Fréchet distance was first proposed by Godau [86]. Guibas *et al.* [92] gave an $O\left(m^2 \log^2 m\right)$ time algorithm for computing the minimum-size weak simplification under the continuous Fréchet distance in $\mathbb{R}^2$. Such a simplification assumes that the vertices of the simplification curve do not necessarily come from the original curve. The current state-of-the-art approximation algorithm for (weak) simplification under the Fréchet distance was suggested by Agarwal *et al.* [7]. This algorithm computes a 2-approximate minimal-size simplification in time $O(m \log m)$. They also gave a simplification that has at most the complexity of an optimal simplification with an 8-approximation of the error.

Recently, van de Kerkhof *et al.* [171] showed that a weak simplification with twice the complexity of an optimal solution and a $(1+\varepsilon)$-approximation of the error can be found in time $O\left(m^2 \log m \log\log m\right)$. For the problem of a vertex-restricted simplification under continuous Fréchet distance, Bringmann and Chaudhury [32] gave an $O\left(m^3\right)$ time solution, and simultaneously, a conditional cubic lower bound. This improved a previous polynomial time algorithm of van Kreveld, Löffler and Wiratma [172].

Driemel and Har-Peled [65] considered the computing of the Fréchet distance with shortcuts, i.e. with local simplifications. They showed how to preprocess a polygonal curve in near-linear time and space, and introduced the concept of a *vertex permutation*, such that, intuitively, any prefix of this permutation represents a bicriteria approximation to the minimal-error curve simplification, with respect to the continuous Fréchet distance. In Section 3.4 we will use this concept to develop an efficient algorithm for our simplification curves.

Assuming c-packedness of a curve and that the shortcuts start and end in the vertices of the curve, Driemel and Har-Peled [65] gave a near-linear time $(3+\varepsilon)$-approximation algorithm. A more general variant of the problem where the shortcuts can be taken at any point of the curve is **NP**-hard, as shown by Buchin, Driemel and Speckmann [47]. They also gave a 3-approximation for the decision variant of the problem, in $O\left(m^3 \log m\right)$ time.

Bereg *et al.* [26] gave an $O\left(m \log m\right)$ time algorithm for the minimum-size ε-simplification problem, and a $O\left(m\ell \log m \log(m/\ell)\right)$ for the minimum-error ℓ-simplification problem under the discrete Fréchet distance. Confer to Lemma 4.9 where the latter result is formally stated.

For two given curves, simplifying them independently does not necessarily preserve their resemblance. Bereg *et al.* [26] considered the problem of two simultaneously simplified curves, using vertices of the original curves. This problem is called *chain pair simplification*,

and its two variants are studied: CPS-2H - where the error of the simplifications to the original curves is measured by the Hausdorff distance, and the distance between two simplifications is measured by the discrete Fréchet distance; and CPS-3F, where all three distances are discrete Fréchet distances. They showed that the CPS-2H problem is **NP**-complete, and hypothesized the same for the CPS-3F problem.

Wylie and Zhu [176] considered the protein chain pair simplification under the discrete Fréchet distance. The polygonal curve (called *chain*) alignment and comparison with respect to the proteins is a central problem in structural biology. Proteins' alignment was previously studied using root mean square deviation (RMSD). The discrete Fréchet distance is a natural choice option for alignment of the vertices, that represent carbon atoms in proteins. On contrary, the continuous Fréchet distance would map arbitrary points on the curves, which is biologically not meaningful. They gave a 2-approximation algorithm to the CPS-3F problem.

Fan *et al.* [78] showed that the chain pair simplification (CPS-3F) problem is polynomially solvable – in time $O\left(m^5\right)$. The general chain pair simplification problem, when the vertices of the simplification curve are not limited to the vertices of the original curves, was considered by Fan *et al.* [79], who gave both an exact algorithm with $\tilde{O}(m^7)$ running time, and an approximation algorithm, with $O\left(m^4\right)$ running time.

In reality, the curves or even their complexity do not have to be known completely in advance. The streaming scenario describes the case when the vertices of the input curve are presented one at a time. The curve simplification algorithm under the Fréchet distance in a streaming fashion was first studied by Abam *et al.* [1]. They considered simplification of general paths in $\mathbb{R}^2$ under the Fréchet distance, using the framework of Imai and Iri [107]. The authors of [1] obtained an algorithm that, for any fixed $\varepsilon > 0$, produces $(4\sqrt{2}+\varepsilon)$-competitive streaming algorithm, that uses $O(\ell^2/\sqrt{\varepsilon})$ additional storage and processes each input point in $O(\ell \log(1/\varepsilon))$ amortized time. Their algorithm allows resource augmentation, more precisely, it uses a 2ℓ-simplification, but compares its error to the optimal ℓ-simplification. A similar assumption was previously used by Agarwal *et al.* [7].

The result of Abam *et al.* [1] was improved by Driemel, Psarros and Schmidt [70], who obtained an 8-approximation to the optimal ℓ-simplification of the input curve from $\mathbb{R}^d$ in the streaming scenario under the discrete Fréchet distance, and without resource augmentation. Their algorithm needs time $O\left(d\ell\right)$ per update, and uses space $O\left(d\ell\right)$. Recently, Filtser and Filtser [83] gave, based on a minimal enclosing ball streaming algorithm, a $(1.22+\varepsilon)$-approximation algorithm to the optimal ℓ-simplification in a streaming fashion, using space that is linear in both ℓ and d, and almost linear dependency on $1/\varepsilon$. Additionally, they

gave a $(1+\varepsilon)$-approximation to the optimal ℓ-simplification in a streaming fashion, but at the cost of additional $(1/\varepsilon)^{O(d)}$ space required.

Recently, our δ-signatures have found the first application area outside of our work ([68]), where they have been originally introduced. Driemel and Psarros [69] used signatures to construct a $(2+\varepsilon)$-approximation to the approximate near neighbor problem for the one-dimensional polygonal curves.

3.2 On properties of signatures

In this chapter, for given two curves τ and π, we analyze a matching of these curves, i.e. the function that realizes the Fréchet distance $d_F(\tau,\pi)$. By the definition of the (continuous) Fréchet distance in Equation (2.24), such a matching f is a bijection between parametrizations of τ and π. The bijection that realizes $d_F(\tau,\pi)$ does not always exist, but it is obtained only in limit, since in the definition there is *infimum* instead of *minimum*. In the literature it is well-known and commonly used[9], that a proper bijection can be obtained by a slight perturbation, such that for any $\varepsilon > 0$ this bijection realizes the Fréchet distance $d_F(\tau,\pi)+\varepsilon$. Without loss of generality we will thus construct matchings which are not bijections, but can be perturbed into one.

In this section we prove several useful properties of the signatures. These properties will be crucial for the application of signatures to the clustering of curves in one-dimensional ambient space. First we show that a δ-signature σ of a given curve τ is indeed a curve simplification with bounded error, which approximates the original curve well, since its Fréchet distance to τ is at most δ. This is claimed by Lemma 3.4.

Lemma 3.4. *It holds for any δ-signature σ of τ that $d_F(\tau,\sigma) \leq \delta$.*

Proof. Let $0 = s_1 < \ldots < s_\ell = 1$ be the series of parameter values of vertices on τ that describe σ. We construct a matching between each signature edge $e_i = \overline{\tau(s_i)\tau(s_{i+1})}$, $1 \leq i < \ell$, and the corresponding subcurve $\widehat{\tau}_i = \tau[s_i, s_{i+1}]$ of τ, in a greedy manner.

Assume first, for simplicity, that it holds $\tau(s_i) < \tau(s_{i+1})$ (i.e. the edge of the signature is directed upwards at the time) and none of its endpoints are endpoints of τ. We process the vertices w_j of the subcurve $\widehat{\tau}_i$ while keeping a current position v on the edge e. The idea is to walk as far as possible on $\widehat{\tau}$ while walking as little as possible on e_i. We initialize $v = \tau(s_i)$, and match the first vertex $w_i = \tau(s_i)$ of $\widehat{\tau}_i$ to v (i.e. to itself). The invariant that each vertex $w \in \widehat{\tau}_i$ is at distance at most δ to the matched point v is satisfied. When processing a vertex $w_j \in \widehat{\tau}_i$, we update v to $\max(v, w_j - \delta)$, and match w_j to the current

[9] A proof can be found in the paper by Buchin, Driemel and Rohde [46].

position v on e_i. The invariant remains valid by induction over vertices w_j and by the direction-preserving property in Definition 3.3. At the end we match $\tau(s_{i+1})$ to itself. By Lemma 2.37, every subcurve of $\widehat{\tau}_i$ is matched to a subsegment of e_i within Fréchet distance δ. Lemma 2.36 implies that $d_F(\widehat{\tau}_i, e_i) \leq \delta$.

If, for the edge e_i, it holds that $\tau(s_i) > \tau(s_{i+1})$ (edge directed downwards), the construction can be done symmetrically by walking backwards on $\widehat{\tau}_i$ and e_i. If the first vertex w_i of $\widehat{\tau}_i$ is $\tau(0)$, we start the construction above with the first vertex w_j, that lies outside the range $[\tau(0) - \delta, \tau(0) + \delta]$. The skipped vertices (from w_i to w_j) can be matched to $\tau(0)$, being all at distance at most δ. In the remaining case: if the last vertex of $\widehat{\tau}_i$ is an endpoint of τ, we can again walk backwards on $\widehat{\tau}$ and e_i from $\tau(1)$, and the case is analogous to the case of $\tau(0)$.

Joining all edges of σ, and all subcurves of τ, by Lemma 2.36 we have that $d_F(\tau, \sigma) \leq \delta$, as claimed. □

A signature does not only provide a good approximation to the original curve, but also provides a description for all the curves that are similar to the original curve. That is, any curve that is close (i.e. at small Fréchet distance) to the curve τ has to have vertices close to the vertices of the signature. This is formally stated by Lemma 3.5.

Lemma 3.5. *Let $\sigma = v_1, \ldots, v_\ell$ be a δ-signature of $\tau = w_1, \ldots, w_m$. Let $R_i = [v_i - \delta, v_i + \delta]$, for $1 \leq i \leq \ell$, be ranges centered at the vertices of σ ordered along σ. It holds for any curve π that if $d_F(\tau, \pi) \leq \delta$, then π has a vertex in each range R_i, and such that these vertices appear on π in the order of i.*

Proof. For any $i = \{3, \ldots, \ell - 2\}$, the vertices v_{i-1}, v_i and v_{i+1} satisfy that $|v_i - v_{i-1}| > 2\delta$ and $|v_{i+1} - v_i| > 2\delta$, by the minimum-edge-length property. This implies that $R_{i-1} \cap R_i = \emptyset$ and $R_i \cap R_{i+1} = \emptyset$. Let $\pi(p_i)$ be the point matched to v_i under a matching that witnesses $d_F(\tau, \pi)$, for all $1 \leq i \leq \ell$. It holds that $0 = p_1 < p_2 < \ldots < p_\ell = 1$. Therefore, the curve π visits the ranges R_{i-1}, R_i, and R_{i+1} in the order of the index i.

Since $v_i \notin \langle\langle v_{i-1}, v_{i+1} \rangle\rangle$ the curve π must change direction (from increasing to decreasing or vice versa) between visiting R_{i-1} and R_{i+1}. Furthermore, π cannot go beyond R_i between visiting R_{i-1} and R_{i+1}, i.e. there is no point $x \in \pi\,[p_{i-1}, p_{i+1}]$ such that it holds that $x \notin R_i$ and there is an ordering $v_{i-1} < v_i < v_i + \delta < x$ or $v_{i-1} > v_i > v_i - \delta > x$. This follows from v_i being a local extremum on τ. Therefore, the change of the direction of π takes place in a vertex in R_i.

For $i = 2$ we use a similar argument. Note that $\pi(0)$ has to be matched to v_1 by the definition of the Fréchet distance. As before, π has to visit the ranges R_2 and R_3 in this order and it holds that $R_2 \cap R_3 = \emptyset$. Either the first vertex of π already lies in R_2, or π has

to change direction again, and therefore needs to have a vertex in R_2. The case $i = \ell - 1$ is symmetric. The fact that the points $\tau(0)$ and $\tau(1)$ have to be matched to $\pi(0)$ and $\pi(1)$, respectively, closes the proof. □

The following corollary is a direct implication of Lemma 3.5 and the minimum-edge-length property in Definition 3.3, since σ is a δ-signature and there has to be at least one vertex in each of the ranges centered in vertices which are not endpoints of τ.

Corollary 3.6. *Let σ be a signature of τ with ℓ vertices and $d_F(\sigma, \tau) \leq \delta$. Then any curve π with $d_F(\pi, \tau) \leq \delta$ needs to have at least $\ell - 2$ vertices.*

In order to prove the main result of this section we need to prove the following lemma, which is a slight variation of the main result (Theorem 3.8) and which simplifies the case when the Fréchet distance is obtained in the limit. Its proof is, as said in the introduction to this chapter, quite long, and we separate it into the next Section 3.3.

Lemma 3.7. *Let $\sigma = v_1, \ldots, v_\ell$ be a δ-signature of $\tau = w_1, \ldots, w_m$. Let R_j, $1 \leq j \leq \ell$, be ranges centered at the vertices of σ ordered along σ, where $R_1 = [v_1 - 4\delta, v_1 + 4\delta]$, $R_\ell = [v_\ell - 4\delta, v_\ell + 4\delta]$, and $R_j = [v_j - \delta, v_j + \delta]$ for $2 \leq j \leq \ell - 1$. Let π be a curve with $d_F(\tau, \pi) < \delta$, and let π' be a curve obtained by removing some vertex $u_i = \pi(p_i)$ from π with $u_i \notin \bigcup_{1 \leq j \leq \ell} R_j$. For any $\varepsilon > 0$, it holds that $d_F(\tau, \pi') \leq \delta + \varepsilon$.*

From Lemma 3.7 we obtain Theorem 3.8 – our main result of Section 3.2. Intuitively it states that for a given curve τ and its signature σ, a curve π that is close to τ can be adapted by omitting vertices which are far from the vertices of the signature σ. By doing such an adaptation, the distance of the adapted curve to the curve τ will remain bounded by the distance the curve had before the adaptation.

Theorem 3.8. *Let $\sigma = v_1, \ldots, v_\ell$ be a δ-signature of $\tau = w_1, \ldots, w_m$. Let R_j, $1 \leq j \leq \ell$, be ranges centered at the vertices of σ ordered along σ, where $R_1 = [v_1 - 4\delta, v_1 + 4\delta]$, $R_\ell = [v_\ell - 4\delta, v_\ell + 4\delta]$, and $R_j = [v_j - \delta, v_j + \delta]$ for $2 \leq j \leq \ell - 1$. Let π be a curve with $d_F(\tau, \pi) \leq \delta$, and let π' be a curve obtained by removing some vertex $u_i = \pi(p_i)$ from π with $u_i \notin \bigcup_{1 \leq j \leq \ell} R_j$. It holds that $d_F(\tau, \pi') \leq \delta$.*

Proof. Given are curves τ and π, and a parameter $\delta > 0$, such that $d_F(\tau, \pi) \leq \delta$. By the definition of the Fréchet distance it holds for any $\varepsilon > 0$ that $d_F(\tau, \pi) < \delta + \varepsilon$. Let $\delta' = \delta + \varepsilon$ for some $\varepsilon > 0$ small enough such that:

(i) the δ-signature of τ is equal to the δ'-signature of τ. Such a signature always exists, since there is an arbitrarily small $\varepsilon > 0$, such that δ- and δ'-signatures are equal. This is shown by Lemma 3.34 in Section 3.4.

(ii) any vertex u_i of π satisfying the conditions in Theorem 3.8 also satisfies the conditions of Lemma 3.7 for δ'.

Now we can apply Lemma 3.7 using δ': let π be a curve with $d_F(\tau, \pi) < \delta'$, then for the curve π' it is $d_F(\tau, \pi') \leq \delta' + \varepsilon = \delta + 2\varepsilon$. Since this is implied for any $\varepsilon > 0$ small enough, we have $d_F(\tau, \pi') = \lim_{\varepsilon \to 0} d_F(\tau, \pi') \leq \lim_{\varepsilon \to 0} (\delta + 2\varepsilon) = \delta$, as claimed. □

Theorem 3.8 implies that a search for vertices of a curve, which is close to the curve τ, can be done in areas close to the signature vertices, while the rest can be ignored. But first we need to prove Lemma 3.7.

3.3 The proof of Lemma 3.7

Let f denote the matching from π to τ that witnesses $d_F(\tau, \pi)$. It maps each point on π to a point on τ within distance δ. Such a matching f exists since $d_F(\tau, \pi) < \delta$. Intuitively, we removed u_i and its incident edges from π by replacing the incident edges with a new "edge" connecting the two subcurves that were disconnected by the edge removal. The obtained curve is called π'. We want to construct a matching f' from π' to τ, based on f to show that their Fréchet distance is at most $\delta + \varepsilon$. We are actually going to construct a mapping f' between π' and τ that is not a bijection, but continuous and monotonically non-decreasing. However, this mapping f' will be a bijection between the respective subcurves of π' and τ. Each of these bijections will witness the Fréchet distance at most δ. Then, it is well known that f' can be slightly perturbed to become a bijection between π' and τ, i.e. a matching that witnesses the distance at most $\delta + \varepsilon$ for arbitrarily small $\varepsilon > 0$.

Because of the continuity of the curves, we have to formally describe the "edge" connecting disconnected parts. Let $\pi(p_{i-1})$ and $\pi(p_{i+1})$ be the endpoints of the disconnected components. Let $\pi[p^-, p^+]$ denote the subcurve by which π and π' differ. We call this subcurve a **missing part**. In particular, p^- and p^+ are such that π' can be written as a concatenation of a prefix and a suffix curve of π: $\pi' = \pi[0, p^-] \oplus \pi[p^+, 1]$ and p_i is contained in the open interval (p^-, p^+). Note that $\pi(p^-) = \pi(p^+)$ (i.e. it is the same point, with two different time stamps). Furthermore, it is clear that $\pi[p^-, p^+]$ consists of two edges with u_i being the minimum or maximum connecting them, since we work in the one-dimensional ambient space. If u_i would have been neither a minimum nor a maximum on π, then $\pi[p^-, p^+]$ would be empty. In this case the claim would hold trivially.

The new "edge" $\pi'[p_{i-1}, p_{i+1}]$ consists of three parts: the edge $\pi[p_{i-1}, p^-]$, the point $\pi[p^-]$ and the edge $\pi[p^+, p_{i+1}]$. This is illustrated by Figure 3.2.

In the construction of f' we need to show that the subcurve $\tau[f(p^-), f(p^+)]$, which we call a **broken part**, and which was matched by f^{-1} to the missing part, can be matched

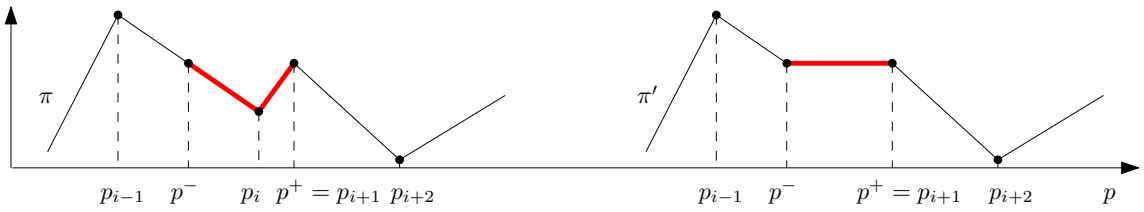


Figure 3.2: The removal of the vertex $\pi(p_i)$ from π. The curves $\pi[p^-, p^+]$ and $\pi'[p^-, p^+]$ are marked red

to some subcurve of π', while respecting the monotonicity of the matching. The proof is a case analysis based on the structure of the two curves. The structure of the proof is roughly as follows: we consider first the trivial case (Case 1), where the missing part matching can be repaired by a point. Then we introduce some basic tools to bound the extent of the repairing of the missing part. This is done in Subsection 3.3.1. Once these tools are established, based on behavior of the curve π we analyze three non-trivial cases in Subsection 3.3.2. However, there still remains Case 5, where an iterative scheme occurs on π, but it can be solved using the techniques from the three previously analyzed non-trivial cases. This is done in Subsection 3.3.3. Finally (in Subsection 3.3.4), we consider the case of the first/last signature edge. That is, the removed vertex u_i was matched by f to a point on the subcurve of τ, whose endpoints are either the first two or the last two vertices of the δ-signature of τ.

In order to focus on the essential arguments, we first make some global assumptions stated below. The first two assumptions can be made without loss of generality. We also introduce some basic notation which is used throughout the rest of the proof.

Assumption 1. *We assume that $\pi(p_i)$ is a local minimum on π.*

If Assumption 1 would not hold, we could first mirror the curves τ and π across the horizontal time axis to obtain the property of Assumption 1 without changing the Fréchet distance.

Let the minimum point on the broken part of the curve τ be denoted

$$z_{\min} = \operatorname*{arg\,min}_{t \in [f(p^-), f(p^+)]} \tau(t).$$

Let $\tau[s_j, s_{j+1}]$ be the subcurve of τ bounded by two consecutive signature vertices, such that $z_{\min} \in [s_j, s_{j+1}]$.[10]

Assumption 2. *We assume that $\tau(s_j) < \tau(s_{j+1})$.*

[10] The statement of Lemma 3.7 uses vertices v_i of the signature σ of the curve τ. However, throughout the proof we work with parameters s_i, with $v_i = \sigma(s_i) = \tau(s_i)$ (see page 49).

If Assumption 2 would not hold, we could first reparametrize the curves τ and π with a new parametrization $\phi(t) = 1 - t$, i.e., we reverse the direction of the time axis, to obtain the property of Assumption 2 without changing the Fréchet distance. Note that this does not interfere with Assumption 1, i.e. does not change the property of $\pi(p_i)$ being a local minimum.

Assumption 3. *We assume that neither $s_j = 0$, nor $s_{j+1} = 1$.*

The cases omitted by Assumption 3 are boundary cases, that will be handled at the end of the proof in Subsection 3.3.4.

For simplicity of presentation, we state the characteristics of the curve τ and its δ-signature σ, that we obtain from Definition 3.3 under Assumption 2, in the following property.

Property 1 (**Signature**). *We can assume that*

(i) $\tau(s_{j+1}) - \tau(s_j) > 2\delta$;
(ii) $\tau(s_j) = \min(\tau[s_{j-1}, s_{j+1}])$;
(iii) $\tau(s_{j+1}) = \max(\tau[s_j, s_{j+2}])$;
(iv) $\tau(t') - 2\delta \leq \tau(t'')$ *for* $s_j \leq t' < t'' \leq s_{j+1}$;
(v) $\tau(s_{j+1}) - \tau(s_{j+2}) > 2\delta$.

By the general position assumption, the minimum $\tau(s_j)$ and the maximum $\tau(s_{j+1})$ are unique on their respective subcurves.

From the condition that $d_F(\tau, \pi) < \delta$ we have the following property.

Property 2 (**Fréchet**). *Any two points matched by f have distance at most δ from each other. In particular, for any two $0 \leq p < p' \leq 1$, it holds that*

(i) $\tau(f(p)) - \delta \leq \pi(p) \leq \tau(f(p)) + \delta$,
(ii) $\min(\tau[f(p), f(p')]) - \delta \leq \min(\pi[p, p']) \leq \min(\tau[f(p), f(p')]) + \delta$,
(iii) $\max(\tau[f(p), f(p')]) - \delta \leq \max(\pi[p, p']) \leq \max(\tau[f(p), f(p')]) + \delta$.

Our proof is structured as case analysis. We consider first the case $\tau(z_{\min}) \geq \pi(p^-) - \delta$. This is illustrated by Figure 3.3. In this case, the whole broken part of τ can be matched to a point.

Case 1 (**Trivial case**). $\tau(z_{\min}) \geq \pi(p^-) - \delta$

Claim 3.9 (**Correctness of Case 1**). *If the conditions of Case 1 are satisfied, then it is $d_F(\pi', \tau) \leq \delta$.*

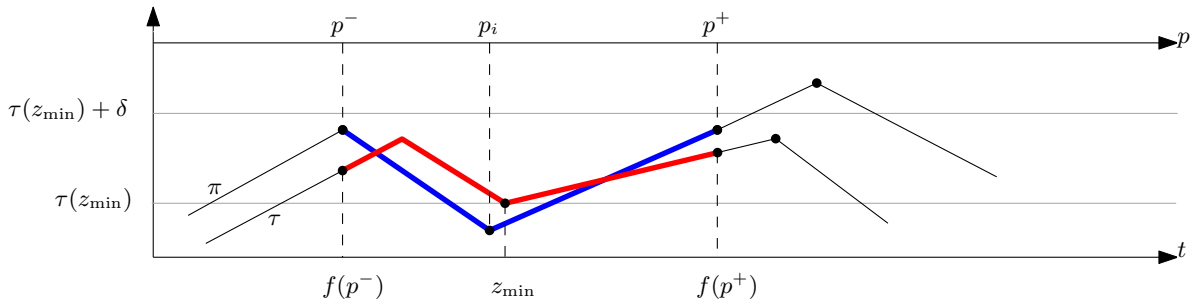


Figure 3.3: Example of Case 1. The broken part of the matching f (the broken part of τ and the missing part of π) is indicated by thick colored lines.

Proof. In this case, we can simply match $\pi(p^-)$ to the subcurve $\tau[f(p^-), f(p^+)]$ and the remaining subcurves $\pi[0, p^-]$ and $\pi[p^+, 1]$ can be matched to the respective subcurves of τ, as it was done by f. By the case distinction it is

$$\min(\tau[f(p^-), f(p^+)]) = \tau(z_{\min}) \geq \pi(p^-) - \delta, \tag{3.1}$$

and by Property 2(iii) it is

$$\max(\tau[f(p^-), f(p^+)]) \leq \max(\pi[p^-, p^+]) + \delta = \pi(p^-) + \delta. \tag{3.2}$$

From Equations (3.1) and (3.2) follows that

$$\{\tau(t) \,:\, t \in [f(p^-), f(p^+)]\} \subseteq [\pi(p^-)]_\delta.$$

Thus, by Lemma 2.36 it holds in this case that $d_F(\pi', \tau) \leq \delta$, as claimed. □

For the rest of the proof we can make the following assumption.

Assumption 4 (Non-trivial case). *We assume that* $\tau(z_{\min}) < \pi(p^-) - \delta$.

3.3.1 Bounds on the matching

Intuitively, we want to extend the subcurves of the trivial case in order to fix the broken matching. The difficulty lies in finding suitable subcurves which cover the broken part $\tau[f(p^-), f(p^+)]$ and whose Fréchet distance is at most δ. Furthermore, the endpoints need to line up appropriately such that we can re-use f for the suffix and the prefix curve $\tau[f(0), f(p^-)]$ and $\tau[f(p^+), f(1)]$, respectively.

The next two claims settle the question to which extent signature vertices $\tau(s_j)$, $\tau(s_{j+1})$, and $\tau(s_{j+2})$ can be included in the subcurve $\tau[f(p^-), f(p^+)]$ for which we need to fix the

broken matching. It holds that only the vertex $\tau(s_{j+1})$ can belong to the subcurve, the other two vertices cannot.

Claim 3.10. *If $s_{j+1} \in [f(p^-), f(p^+)]$ then*

$$\big\{\tau(t) \,:\, t \in [s_{j+1}, f(p^+)]\big\} \subseteq [\tau(s_{j+1}) - 2\delta, \tau(s_{j+1})].$$

Furthermore, it is $s_{j+2} \notin [f(p^-), f(p^+)]$.

Proof. We have to prove that

$$\min(\tau[s_{j+1}, f(p^+)]) \geq \tau(s_{j+1}) - 2\delta \text{ and } \max(\tau[s_{j+1}, f(p^+)]) \leq \tau(s_{j+1}).$$

The subcurve $\pi[p^-, p^+]$ consists of two edges $\overline{\pi(p^-)\pi(p_i)}$ and $\overline{\pi(p_i)\pi(p^+)}$, where $\pi(p_i)$ is the minimum of the subcurve. For the lower bound we distinguish two cases: $p^- \leq f^{-1}(s_{j+1}) < p_i$ and $p_i \leq f^{-1}(s_{j+1}) \leq p^+$.

If $p^- \leq f^{-1}(s_{j+1}) < p_i$, then since $z_{\min} < s_{j+1}$, the subcurve $\pi[p^-, f^{-1}(s_{j+1})]$ is decreasing (by Assumption 1). Thus, by Property 2 it holds that

$$\begin{aligned}\tau(z_{\min}) = \min(\tau[f(p^-), s_{j+1}]) &\geq \min(\pi[p^-, f^{-1}(s_{j+1})]) - \delta = \pi(f^{-1}(s_{j+1})) - \delta \\ &\geq \tau(s_{j+1}) - 2\delta. \end{aligned} \tag{3.3}$$

From the definition of $z_{\min}$, it follows that

$$\min(\tau[s_{j+1}, f(p^+)]) \geq \min(\tau[f(p^-), f(p^+)]) = \tau(z_{\min}) \overset{(3.3)}{\geq} \tau(s_{j+1}) - 2\delta.$$

If $p_i \leq f^{-1}(s_{j+1}) \leq p^+$, the subcurve $\pi[p^-, f^{-1}(s_{j+1})]$ is increasing. Then by Property 2

$$\min(\tau[s_{j+1}, f(p^+)]) \geq \min(\pi[f^{-1}(s_{j+1}), p^+]) - \delta = \pi(f^{-1}(s_{j+1})) - \delta \geq \tau(s_{j+1}) - 2\delta,$$

and the lower bound on $\min(\tau[s_{j+1}, f(p^+)]$ is proved.

Furthermore, by Property 1(v) it follows from the lower bound on $\min(\tau[s_{j+1}, f(p^+)]$, that

$$\min(\tau[s_{j+1}, f(p^+)]) \geq \tau(s_{j+1}) - 2\delta > \tau(s_{j+2}),$$

and therefore $s_{j+2} \notin [f(p^-), f(p^+)]$.

The upper bound follows from Property 1(iii):

$$\max(\tau[s_{j+1}, f(p^+)]) \leq \max(\tau[s_{j+1}, s_{j+2}]) = \tau(s_{j+1}).$$

This closes the proof. □

Claim 3.11. *It holds that* $s_j \notin [f(p^-), f(p^+)]$.

Proof. For the sake of contradiction, assume the claim is false, i.e. $s_j \in [f(p^-), f(p^+)]$. We have (by definition)

$$z_{\min} \in [f(p^-), f(p^+)] \cap [s_j, s_{j+1}].$$

Furthermore, by definition $\tau(z_{\min}) = \min(\tau[f(p^-), f(p^+)])$, and by Property 1(ii), we have $\tau(s_j) = \min(\tau[s_{j-1}, s_{j+1}])$. The assumption $s_j \in [f(p^-), f(p^+)]$ would imply that

$$\tau(z_{\min}) = \min\left\{\tau(t)\colon t \in [f(p^-), f(p^+)] \cap [s_j, s_{j+1}]\right\} = \tau(s_j).$$

By the lemma statement is $\pi(p_i) \notin [\tau(s_j)]_\delta = [\tau(z_{\min})]_\delta$. However, by Property 2, we have

$$\pi(p_i) = \min(\pi[p^-, p^+]) \in [\min(\tau[f(p^-), f(p^+)])]_\delta = [\tau(z_{\min})]_\delta = [\tau(s_j)]_\delta,$$

a contradiction. □

We now introduce some additional notation which will be used throughout the proof. Let

$$\begin{aligned} t_{\min} &= \operatorname*{arg\,min}_{t \in [f(p^-), s_{j+1}]} \tau(t), \\ x &= \max\{p \in [0, p^-]\colon \pi(p) = \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\}\}, \\ p_{\max} &= \operatorname*{arg\,max}_{p \in [x, p^-]} \pi(p), \\ y &= \min\{t \in [t_{\min}, 1]\colon \tau(t) = \pi(p_{\max}) - \delta\}. \end{aligned}$$

In the next few claims we argue that these variables are well-defined. In particular, that x and y always exist in the non-trivial case (Claim 3.12 and Claim 3.13, respectively). Clearly, $t_{\min}$ is well-defined. It holds that $z_{\min} \leq t_{\min}$. To see this, we observe two cases on s_{j+1}.

$s_{j+1} \in [f(p^-), f(p^+)]$: By the definition we have that $z_{\min} \leq s_{j+1}$, and thus $z_{\min} \in [f(p^-), s_{j+1}]$. This implies that $z_{\min} = t_{\min}$.

$s_{j+1} \notin [f(p^-), f(p^+)]$: It is either $t_{\min} \in [f(p^-), f(p^+)]$, implying $z_{\min} = t_{\min}$, or $t_{\min} \in [f(p^+), s_{j+1}]$, in which case we have $z_{\min} < t_{\min}$.

We also derive some bounds along the way, which will be used throughout the later parts of the proof.

Claim 3.12 (**Existence of** x)**.** *It holds that*

(i) $\min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\} \in \{\pi(p) \colon p \in [f^{-1}(s_j), p^-]\}$;
(ii) $\min(\pi[x, p^-]) \geq \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\} = \pi(x)$;
(iii) $\tau(s_j) \leq \tau(t_{\min})$.

Proof. We first prove part (i) of the claim. We show that there exist two parameters p_1 and p_2, with $f^{-1}(s_j) \leq p_1 \leq p_2 \leq p^-$, such that

$$\pi(p_1) \leq \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\} \leq \pi(p_2). \tag{3.4}$$

Since the curve π is continuous, this would imply the claim, and also imply the existence of x. Indeed, we can choose $p_1 = f^{-1}(s_j)$ and $p_2 = p^-$. If $s_{j+1} \geq f(p^+)$, we have

$$\pi(p_2) = \pi(p^-) > \tau(z_{\min}) + \delta \geq \tau(t_{\min}) + \delta, \tag{3.5}$$

since we assume the non-trivial case $\tau(z_{\min}) < \pi(p^-) - \delta$. Note that the last inequality in Equation (3.5) holds independently of the case assumption $s_{j+1} \geq f(p^+)$. Otherwise, if $s_{j+1} < f(p^+)$, it is $\tau(s_{j+1}) = \max(\tau[f(p^-), f(p^+)])$ by Property 1(iii), since $[f(p^-), f(p^+)] \subseteq [s_j, s_{j+2}]$. Then by Property 1 and Property 2, we have that

$$\pi(p_2) = \pi(p^-) = \max(\pi[p^-, p^+]) \geq \max(\tau[f(p^-), f(p^+)]) - \delta = \tau(s_{j+1}) - \delta.$$

Thus, in both cases, it holds that $\pi(p_2) \geq \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\}$.

As for p_1, by Claim 3.11 and the definition of $t_{\min}$, we have $0 \leq s_j \leq f(p^-) \leq t_{\min} \leq s_{j+1}$. By Property 2(i), it is

$$\pi(p_1) = \pi(f^{-1}(s_j)) \leq \tau(s_j) + \delta.$$

It follows by Property 1(ii) that $\pi(p_1) \leq \min(\tau[s_{j-1}, s_{j+1}]) + \delta = \tau(t_{\min}) + \delta$, and by Property 1(i) that $\pi(p_1) < \tau(s_{j+1}) - \delta$. Thus, $\pi(p_1) \leq \min\{\tau(s_{j+1}) - \delta, \tau(s_{j+1}) - \delta\}$, and the part (i) of the claim is proved.

The part (ii) of the claim follows directly from Equation (3.4). It is $\min(\pi[x, f(p^-)]) = \pi(x)$, since $\pi(x)$ is defined as the last point along the prefix subcurve $\pi[0, p^-]$ with the specified value, and π is continuous. The part (iii) follows from the Property 1(ii), implying $\tau(s_j) = \min(\tau[s_{j-1}, s_{j+1}])$, and the definition of $\tau(t_{\min}) = \min(\tau[f(p^-), s_{j+1}])$. □

Claim 3.13 (**Existence of** y)**.** *It holds that*

(i) $\pi(p_{\max}) - \delta \in \{\tau(t) \colon t \in [t_{\min}, s_{j+1}]\}$;
(ii) $\max(\tau[t_{\min}, y]) \leq \pi(p_{\max}) - \delta = \tau(y)$;
(iii) $\pi(p_{\max}) \leq \tau(s_{j+1}) + \delta$.

Proof. To prove part (i) of the claim, which will also imply the existence of y, we show that there exist two parameters t_1 and t_2, with $t_{\min} \leq t_1 \leq t_2 \leq s_{j+1}$, such that

$$\tau(t_1) \leq \pi(p_{\max}) - \delta \leq \tau(t_2).$$

We choose $t_1 = t_{\min}$ and $t_2 = s_{j+1}$. Since we have the non-trivial case, we know that

$$\pi(p_{\max}) - \delta \geq \pi(p^-) - \delta > \tau(z_{\min}) \overset{(3.5)}{\geq} \tau(t_{\min}) = \tau(t_1).$$

Now, for t_2, we know that $s_j \overset{(3.4)}{\leq} f(x) \leq f(p_{\max}) \leq f(p^-) \leq s_{j+1}$. By Property 2 and by Property 1(iii) it is

$$\pi(p_{\max}) - \delta \leq \tau(f(p_{\max})) \leq \tau(s_{j+1}) = \tau(t_2). \tag{3.6}$$

Since the subcurve is continuous, there must be a parameter $t_1 \leq t \leq t_2$ which satisfies the claim, and the part (i) is proved. The part (ii) of the claim also follows directly, since $\tau(y)$ is the first point along the suffix subcurve $\tau[t_{\min}, 1]$ with the specified value, and since $\tau(y) \geq \tau(t_{\min})$. The part (iii) follows from Equation (3.6). □

The following claim follows directly from Claim 3.12, the definitions of x and y, and Claim 3.13, respectively.

Claim 3.14. *It holds that* $s_j \leq f(x) \leq f(p_{\max}) \leq f(p^-) \leq t_{\min} \leq y \leq s_{j+1}$.

The following claim will be used throughout the proof, drawing the relation between $\pi(p_{\max})$ and $\pi(x)$.

Claim 3.15. *It holds that* $\pi(p_{\max}) - 2\delta \leq \pi(x)$.

Proof. We need to show that

$$\pi(p_{\max}) - 2\delta \leq \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\}.$$

Claim 3.13 immediately implies $\pi(p_{\max}) - 2\delta \leq \tau(s_{j+1}) - \delta$. On the other hand, by Claim 3.14 and the definitions of $p_{\max}$ and $t_{\min}$, it is

$$s_j \leq f(x) \leq f(p_{\max}) \leq f(p^-) \leq t_{\min} \leq s_{j+1}.$$

By Property 2 and by Property 1(iv), we have

$$\pi(p_{\max}) - 2\delta \leq \tau(f(p_{\max})) + \delta - 2\delta \leq \tau(t_{\min}) + \delta. \tag{3.7}$$

Thus, the claim is proved. □

The next two claims (Claim 3.16 and Claim 3.17) show that our choice of x and y is suitable for fixing some parts of the broken matching: the subcurve $\pi[x, p^-]$ can be matched entirely to $\tau(y)$ (by Claim 3.16), and the subcurve $\tau[f(x), y]$ can be matched entirely to $\pi(x)$ (by Claim 3.17). After that, it remains to match the subcurve $\pi[p^+, f^{-1}(y)]$. For this, we have the case analysis that follows from Section 3.3.2 on.

Claim 3.16. $\{\pi(p) \,:\, p \in [x, p^-]\} \subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y)]_\delta$.

Proof. By Claim 3.12 and Claim 3.15, respectively, we have that

$$\min(\pi[x, p^-]) \geq \min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\} \geq \pi(p_{\max}) - 2\delta.$$

On the other hand, by the definition of $p_{\max}$, we have $\max(\pi[x, p^-]) = \pi(p_{\max})$. The latter equality of the claim follows directly from the definition of y and from Claim 3.13 (y is well-defined). □

Claim 3.17. $\{\tau(t) \,:\, t \in [f(x), y]\} \subseteq [\min\{\tau(t_{\min}) + \delta, \tau(s_{j+1}) - \delta\}]_\delta = [\pi(x)]_\delta$.

Proof. We first prove the lower bound on the minimum of the subcurve $\tau[f(x), y]$. By Property 2, and by Claim 3.12, we have

$$\min(\tau[f(x), f(p^-)]) \geq \min(\pi[x, p^-]) - \delta \geq \min\{\tau(t_{\min}), \tau(s_{j+1}) - 2\delta\}. \tag{3.8}$$

By definition, $\tau(t_{\min})$ is a minimum on $\tau[f(p^-), s_{j+1}]$. Thus, for $y \leq s_{j+1}$, which is ensured by Claim 3.14, we have that

$$\min(\tau[f(p^-), y]) \geq \min(\tau[f(p^-), s_{j+1}]) = \tau(t_{\min}) \geq \min\{\tau(t_{\min}), \tau(s_{j+1}) - 2\delta\}. \tag{3.9}$$

Equations (3.8) and (3.9) imply the lower bound.

We now prove the upper bound on the maximum of the subcurve $\tau[f(x), y]$. Since by Claim 3.14, $s_j \leq f(x) \leq t_{\min} \leq s_{j+1}$ and, since by Property 1(iv), $\tau[s_j, s_{j+1}]$ may not descend by more than 2δ, it follows that

$$\max(\tau[f(x), t_{\min}]) \leq \tau(t_{\min}) + 2\delta. \tag{3.10}$$

By Claim 3.13, Claim 3.15, and by the definition of x, it is

$$\max(\tau[t_{\min}, y]) \leq \pi(p_{\max}) - \delta \leq \pi(x) + \delta \leq \tau(t_{\min}) + 2\delta. \tag{3.11}$$

For $y \leq s_{j+1}$, which is ensured by Claim 3.14, and by Property 1(iii), we also have that

$$\max(\tau[f(x), y]) \leq \tau(s_{j+1}). \tag{3.12}$$

Equations (3.10), (3.11), and (3.12) together imply the upper bound. The last equality of the claim follows directly from the definition of x and from Claim 3.12 (x is well-defined). □

3.3.2 Non-trivial cases

Now we have established the basic setup for our proof of Lemma 3.7. Since for any $0 \leq p' \leq p^-$, $\pi(p')$ and $\tau(f(p'))$ were already matched by f, witnessing $d_F(\pi, \tau)$, the matching of the prefix curves $\pi[0, p']$ and $\tau[0, f(p')]$ can be reused from f. An analogous claim can be made for the suffix curves. In the rest of the proof we use the notation $\pi[a, b] \Leftrightarrow \tau[c, d]$ to denote that these two subcurves are matched to each other by the new matching (between π' and τ).

In the following, we describe the case analysis based on the structure of the two curves τ and π. Consider walking along the subcurve $\pi[p^+, 1]$. At the beginning of the subcurve, we have $\pi(p^+) \in [\pi(x), \pi(p_{\max})]$. One of the following events may happen during the walk: either we *stay inside this interval*, or *go above* $\pi(p_{\max})$, or we *go below* $\pi(x)$. Let q denote the time at which one of these events occurs for the first time. Formally, we define the intersection function $g : \mathbb{R} \to [p^+, 1] \cup \{p_\infty\}$, as

$$\begin{cases} g(h) &= \min(\{p \in [p^+, 1] \colon \pi(p) = h\} \cup \{p_\infty\}), \\ q &= \min\{g(\pi(p_{\max})), g(\pi(x))\}, \end{cases} \tag{3.13}$$

where $p_\infty > 1$ is some fixed constant for the case that the suffix curve $\pi[p^+, 1]$ does not contain the value h. We distinguish the following main cases. In each of the cases, we devise a matching scheme to fix the broken matching. For each case, our construction ensures that the extended subcurves cover the subcurve $\tau[f(p^-), f(p^+)]$ and that the subcurves line up with suitable prefix and suffix curves, such that we can always use f for the parts of π and τ not covered in the matching scheme. We need to prove in each case that the Fréchet distance between the specified subcurves is at most δ. If this is the case, we call the matching scheme **valid**. By Lemma 2.36 it will follow that $d_F(\pi', \tau) \leq \delta$, and thus, that Lemma 3.7 is correct.

We have to make further distinction between the case when $f(p^+) \leq y$ and the case $f(p^+) > y$. If $f(p^+) \leq y$ holds, the three aforementioned events are described by Case 2, Case 3 and Case 4. If it happens that $f(p^+) > y$, it becomes more complicated to repair the matching. This is discussed in Case 5, which is separated into Subsection 3.3.3.

Case 2 (π **stays level**). $p^+ \leq f^{-1}(y) \leq q$.

Case 2 is the simplest non-trivial case, as the curve π does not reach outside of the interval $[\pi(x), \pi(p_{\max})]$ on the subcurve $\pi[p^+, f^{-1}(y)]$. Confer to Figure 3.4 for an example. We intend to use the following matching scheme:

$$\begin{cases} \pi(x) & \Leftrightarrow & \tau[f(x), y] \\ \pi[x, p^-] & \Leftrightarrow & \tau(y) \\ \pi[p^+, f^{-1}(y)] & \Leftrightarrow & \tau(y), \end{cases} \tag{3.14}$$

and for the suffix curves $\pi[f^{-1}(y), 1]$ and $\tau[y, 1]$ we reuse the matching f.

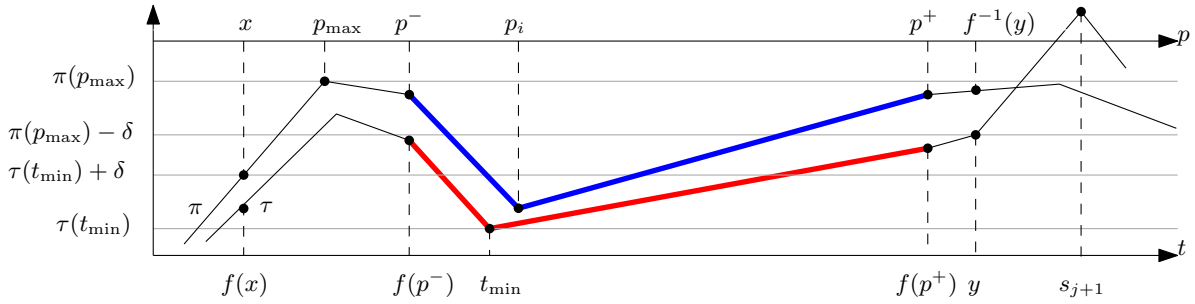


Figure 3.4: Example of Case 2. The broken part of the matching f is indicated by thick lines.

Claim 3.18 (**Correctness of Case 2**). *Let the conditions of Case 2 be satisfied. Then the matching scheme given by Equation (3.14) is valid.*

Proof. Claim 3.17 implies that the Fréchet distance between $\tau[f(x), y]$ and $\pi(x)$ is at most δ. Claim 3.16 implies that the Fréchet distance between $\pi[x, p^-]$ and $\tau(y)$ is at most δ. Finally, by our case distinction and by Claim 3.15 it is

$$\{\pi(p) \,:\, p \in [p^+, f^{-1}(y)]\} \subseteq [\pi(x), \pi(p_{\max})] \subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y)]_\delta.$$

Therefore, also the Fréchet distance between $\pi[p^+, f^{-1}(y)]$ and $\tau(y)$ is at most δ, implying that $d_F(\pi', \tau) \leq \delta$. □

Case 3 (π **tends upwards**). $q < f^{-1}(y)$ *and* $q = g(\pi(p_{\max}))$.

In Case 3, let

$$y' = \max\{t \in [0, f(q)] \colon \tau(t) = \tau(y)\} \quad \text{and} \quad z = \max\{p^+, f^{-1}(y')\}. \tag{3.15}$$

Confer to Figure 3.5 for an example. We intend to use the following matching scheme:

$$\begin{cases} \pi(x) & \Leftrightarrow & \tau[f(x), y'] \\ \pi[x, p^-] & \Leftrightarrow & \tau(y') \\ \pi[p^+, z] & \Leftrightarrow & \tau(y') \\ \pi(z) & \Leftrightarrow & \tau[y', f(p^+)]. \end{cases} \tag{3.16}$$

If $y' > f(p^+)$, then the last line of the matching scheme in (3.16) is simply neglected. Then it is $z = f^{-1}(y')$, and the suffix curves $\pi[f^{-1}(y'), 1]$ and $\tau[y', 1]$ are matched. Otherwise, if $y' \leq f(p^+)$, then $z = p^+$, and we can match $\pi[p^+, 1] \Leftrightarrow \tau[f(p^+), 1]$.

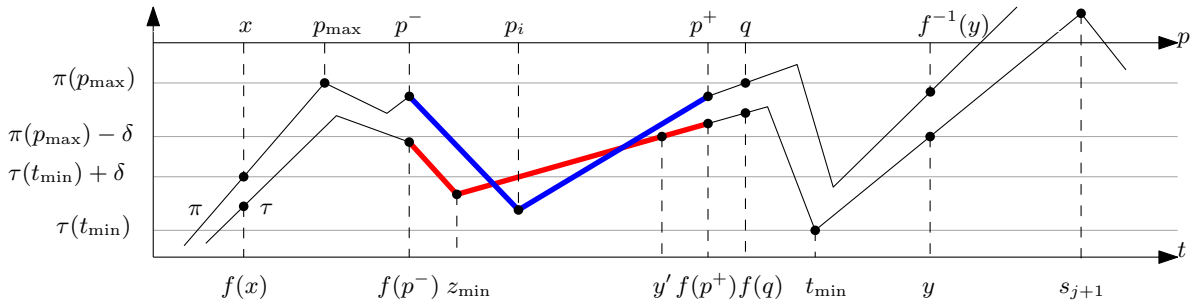


Figure 3.5: Example of Case 3. The broken part of the matching f is indicated by thick lines.

Claim 3.19 (Correctness of Case 3). *Let the conditions of Case 3 be satisfied. Then the matching scheme given by Equation (3.16) is valid.*

Proof. We first argue that y' exists. To do this, we show that there exist two parameters t_1 and t_2, with $0 \leq t_1 < t_2 \leq f(q)$, such that

$$\tau(t_1) \leq \tau(y) = \pi(p_{\max}) - \delta \leq \tau(t_2). \tag{3.17}$$

We choose $t_1 = z_{\min}$ and $t_2 = f(q)$. Such a choice is fine, since by the definitions of $z_{\min}$ and q, it is $z_{\min} \leq f(p^+) \leq f(q)$. Now, by Property 2 and the case distinction,

$$\tau(t_2) = \tau(f(q)) \geq \pi(q) - \delta = \pi(p_{\max}) - \delta.$$

Since we are assuming the non-trivial case,

$$\tau(t_1) = \tau(z_{\min}) < \pi(p^-) - \delta \leq \pi(p_{\max}) - \delta.$$

Thus, since $\tau[0, f(q)]$ is continuous, by Equation (3.17) y' must exist, and it holds that $f(p^-) \leq z_{\min} \overset{(3.17)}{\leq} y'$. It remains to prove that the matching scheme is valid. Since $y' \leq f(q) < y$, Claim 3.17 implies that the Fréchet distance between $\tau[f(x), y']$ and $\pi(x)$ is at most δ. Claim 3.16 implies that the Fréchet distance between $\pi[x, p^-]$ and $\tau(y')$ is at most δ. For the last two lines of the matching scheme we distinguish two subcases:

(i) If $y' > f(p^+)$, then $z = f^{-1}(y')$, and we need to prove that

$$\big\{\pi(p) \,:\, p \in [p^+, f^{-1}(y')]\big\} \subseteq [\tau(y')]_\delta.$$

By Case 3 distinction and by Claim 3.15

$$\big\{\pi(p) \,:\, p \in [p^+, q]\big\} \subseteq [\pi(x), \pi(p_{\max})] \subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y')]_\delta.$$

Since $f^{-1}(y') \leq q$, this implies the validity of the matching.

(ii) If $y' \leq f(p^+)$, then $z = p^+$, and we need to prove that

$$\big\{\tau(t) \,:\, t \in [y', f(p^+)]\big\} \subseteq [\pi(p^+)]_\delta.$$

On the one hand, since $y' \in [f(p^-), f(p^+)]$ by the choice of t_1 and the subcase distinction, we have, by Property 2, that

$$\max(\tau[y', f(p^+)]) \leq \max(\pi[f^{-1}(y'), p^+]) + \delta = \max(\pi[p^-, p^+]) + \delta = \pi(p^+) + \delta.$$

On the other hand, since $y' \leq f(p^+) \leq f(q)$ by the subcase distinction and the definition of q, we have, by the definition of y' as the last point along the prefix subcurve $\tau[0, f(q)]$ with the specified value, that

$$\min(\tau[y', f(p^+)]) = \tau(y') = \pi(p_{\max}) - \delta \geq \pi(p^+) - \delta.$$

Thus, our matching scheme of Case 3 is valid, and it is $d_F(\pi', \tau) \leq \delta$. □

Case 4 (π **tends downwards**). *$q < f^{-1}(y)$ and $q = g(\pi(x))$.*

In Case 4, let

$$y'' = \min\{t \in [f(p_{\max}), 1] \colon \tau(t) = \tau(y)\}.$$

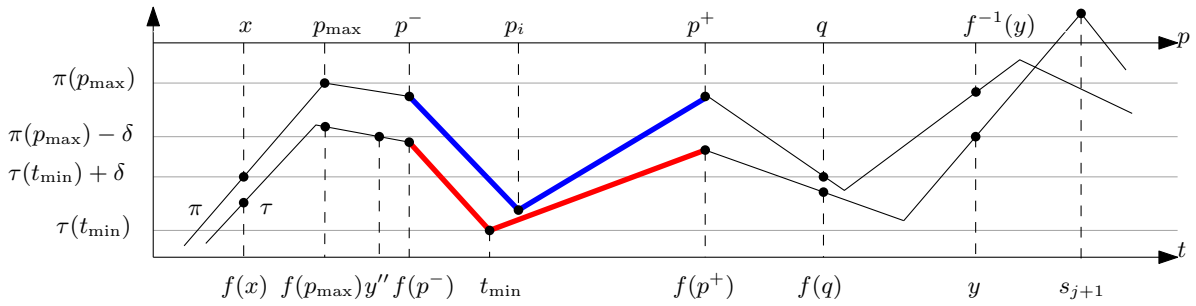


Figure 3.6: Example of Case 4. The broken part of the matching f is indicated by thick lines.

Confer to Figure 3.6 for an example. We intend to use the following matching scheme:

$$\begin{cases} \pi(p_{\max}) & \Leftrightarrow \quad \tau[f(p_{\max}), y''] \\ \pi[p_{\max}, p^-] & \Leftrightarrow \quad \tau(y'') \\ \pi[p^+, q] & \Leftrightarrow \quad \tau(y'') \\ \pi(q) & \Leftrightarrow \quad \tau[y'', f(q)]. \end{cases} \tag{3.18}$$

Since $p_{\max} \leq p^-$ and $q \geq p^+$, we can reuse the matching f to match the respective prefix and the suffix curves.

Claim 3.20 (Correctness of Case 4). *Let the conditions of Case 4 be satisfied. Then the matching scheme given by Equation (3.18) is valid.*

Proof. Clearly, y'' exists in the non-trivial case, since τ is continuous and it is

$$\tau(f(p_{\max})) \geq \tau(y) \geq \tau(z_{\min}) \quad \text{and} \quad f(p_{\max}) \leq f(p^-) \leq z_{\min}.$$

We prove the validity of the matching scheme line by line. Note that by definition $\tau(y'') = \tau(y) = \pi(p_{\max}) - \delta$. For the first matching, by the definition of y'' and by Property 2, we have that $\tau(f(p_{\max})) \geq \tau(y'') = \min(\tau[f(p_{\max}), y'']) = \pi(p_{\max}) - \delta$. On the other hand, it is also $\max(\tau[f(p_{\max}), y'']) = \tau(f(p_{\max})) \leq \pi(p_{\max}) + \delta$. If there would exist $t' \in [f(p_{\max}), y'']$ with $\tau(t') > \pi(p_{\max}) + \delta$, then it would contradict Property 1(iv). Therefore, it is

$$\big\{\tau(t) \,:\, t \in [f(p_{\max}), y'']\big\} \subseteq [\pi(p_{\max})]_\delta.$$

The validity of the second matching follows from Claim 3.16 since $p_{\max} \geq x$. For the third matching, by Case 4 distinction and by Claim 3.15, it is

$$\big\{\pi(t) \,:\, t \in [p^+, q]\big\} \subseteq [\pi(x), \pi(p_{\max})] \subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y'')]_\delta.$$

As for the last matching, since $f(x) \leq f(p_{\max}) \leq y'' \leq y$, and since by Case 4 distinction it is $f(q) < y$, Claim 3.17 implies

$$\big\{\tau(t) \,:\, t \in [y'', f(q)]\big\} \subseteq \{\tau(t) \,:\, t \in [f(x), f(y)]\} \subseteq [\pi(x)]_\delta = [\pi(q)]_\delta.$$

Therefore, our matching scheme is valid, and it is $d_F(\pi', \tau) \leq \delta$. □

3.3.3 The matryoshka case

Up to here we explored the case where $f(p^+) \leq y$, and showed by Claim 3.18, Claim 3.19, and Claim 3.20 that, in this case, Lemma 3.7 is correct. However, if $f(p^+) > y$, the repairing of the broken matching becomes more complicated. We call this case **the matryoshka case**, since it contains an iterative matching scheme, that metaphorically resembles a matryoshka doll.

Case 5 (The matryoshka case). $f(p^+) > y$.

In the previous cases we have already established a suitable set of tools to handle this case. We devise an iterative matching scheme and prove an invariant (stated in Claim 3.21) to verify that the Fréchet distance of the matched subcurves is at most δ. We first define $z^{(1)}_{\min} = z_{\min}$, $t^{(1)}_{\min} = t_{\min}$, $x^{(1)} = x$, and $y^{(1)} = y$. Now, for $a = 2, \ldots$ let

$$\begin{aligned}
z^{(a)}_{\min} &= \operatorname*{arg\,min}_{t \in [y^{(a-1)}, f(p^+)]} \tau(t), \\
t^{(a)}_{\min} &= \operatorname*{arg\,min}_{t \in [y^{(a-1)}, s_{j+1}]} \tau(t), \\
x^{(a)} &= \min\Big\{p \in [x^{(a-1)}, p_{\max}] \colon \pi(p) = \min\{\tau(t^{(a)}_{\min}) + \delta, \tau(s_{j+1}) - \delta\}\Big\}, \\
y^{(a)} &= \min\Big\{t \in [t^{(a)}_{\min}, s_{j+1}] \colon \tau(t) = \pi(p_{\max}) - \delta\Big\}.
\end{aligned}$$

We describe the intended matching scheme. We begin by matching $\pi[0, x^{(1)}] \Leftrightarrow \tau[0, f(x^{(1)})]$ and $\pi(x^{(1)}) \Leftrightarrow \tau[f(x^{(1)}), y^{(1)}]$. The first matching is reused from f, while the validity of the second matching follows from Claim 3.17. We continue with the following subcurves:

$$\begin{cases} \pi[x^{(a-1)}, x^{(a)}] & \Leftrightarrow \quad \tau(y^{(a-1)}) \\ \pi(x^{(a)}) & \Leftrightarrow \quad \tau[y^{(a-1)}, y^{(a)}], \end{cases} \tag{3.19}$$

where the last two matchings are repeated while incrementing a (starting with $a = 2$). We call the part of the matching stated by (3.19) *the iterative part*. After each iteration, we are left with the unmatched subcurves $\pi[x^{(a)}, p^-]$ and $\tau[y^{(a)}, f(p^+)]$.

Case 5(i) (Trivial subcase): It is $f(p^+) > y$, and for some $a \geq 2$, it holds that:

$$\pi(p^-) \leq \tau(z_{\min}^{(a+1)}) + \delta. \tag{3.20}$$

In this case the matching scheme can be completed easily, since this is equivalent to the trivial case (Case 1). We complete the matching with the following scheme

$$\begin{cases} \pi[x^{(a)}, p^-] & \Leftrightarrow \quad \tau(y^{(a)}), \\ \pi(p^+) & \Leftrightarrow \quad \tau[y^{(a)}, f(p^+)]. \end{cases} \tag{3.21}$$

For the suffix curves $\pi[p^+, 1]$ and $\tau[f(p^+), 1]$ we reuse the matching f. In order to prove the correctness of this (sub)case (Case 5(i)), we extend Claim 3.17 as follows. The next claim will be used in the non-trivial (sub)cases as well.

Claim 3.21. *It holds that*

$$\left\{\tau(t) \ : \ t \in [y^{(a-1)}, y^{(a)}]\right\} \subseteq \left[\tau(t_{\min}^{(a)}) \ , \ \min\{\tau(t_{\min}^{(a)}) + 2\delta, \tau(s_{j+1})\}\right] \subseteq [\pi(x^{(a)})]_\delta.$$

Proof. By Claim 3.14 and the definition of $y^{(a)}$, it is $s_j \leq y \leq y^{(a-1)} \leq y^{(a)} \leq s_{j+1}$. By the definition of $t_{\min}^{(a)}$,

$$\min(\tau[y^{(a-1)}, y^{(a)}]) \geq \min(\tau[y^{(a-1)}, s_{j+1}]) = \tau(t_{\min}^{(a)}). \tag{3.22}$$

Let us assume there exists $t' \in [y^{(a-1)}, y^{(a)}]$, such that $\tau(t') - \tau(t_{\min}^{(a)}) > 2\delta$. By the definitions of $y^{(a)}$ and $t_{\min}^{(a)}$, it is $\max(\tau[t_{\min}^{(a)}, y^{(a)}]) = \tau(y^{(a)})$, thus $t' \in [y^{(a-1)}, t_{\min}^{(a)}]$. But this contradicts Property 1(iv), therefore, we have that

$$\max(\tau[y^{(a-1)}, y^{(a)}]) \leq \tau(t_{\min}^{(a)}) + 2\delta. \tag{3.23}$$

By Property 1(iii), we also have that

$$\max(\tau[y^{(a-1)}, y^{(a)}]) \leq \tau(s_{j+1}). \tag{3.24}$$

Equations (3.22), (3.23), and (3.24) prove the first part of the claim. For the second part we use the definition of $\pi(x^{(a)}) = \min\{\tau(t_{\min}^{(a)}) + \delta, \tau(s_{j+1}) - \delta\}$, which implies

$$\begin{aligned} \tau(t_{\min}^{(a)}) &\geq \pi(x^{(a)}) - \delta \\ \min\{\tau(t_{\min}^{(a)}) + 2\delta, \tau(s_{j+1})\} &= \pi(x^{(a)}) + \delta, \end{aligned}$$

as claimed. □

Claim 3.22 (**Correctness of Case 5(i)**). *If for some value of a, $a \geq 2$, it holds that $\pi(p^-) \leq \tau(z_{\min}^{(a+1)}) + \delta$, then the matching scheme given by Equations (3.19) and (3.21) is valid.*

Proof. By Claim 3.16 the first row of (3.19) is valid, since for all values of a it is $\tau(y^{(a)}) = \pi(p_{\max}) - \delta$, and $[x^{(a-1)}, x^{(a)}] \subseteq [x, p^-]$. The second row of (3.19) is valid by Claim 3.21. Thus, the iterative part of the matching scheme is valid.

It remains to prove the validity of the matchings in (3.21). By Claim 3.16, it is

$$\left\{\pi(p) \,:\, p \in [x, p^-]\right\} \subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y)]_\delta = [\tau(y^{(a)})]_\delta.$$

Since $x \leq x^{(a)} \leq p^-$, this implies that the Fréchet distance between $\pi[x^{(a)}, p^-]$ and $\tau(y^{(a)})$ is at most δ. For the second row in (3.21), we have by our case distinction (3.20), that

$$\min(\tau[y^{(a)}, f(p^+)]) = \tau(z_{\min}^{(a+1)}) \geq \pi(p^-) - \delta, \tag{3.25}$$

while (by Property 2) the matching f testifies that

$$\max(\tau[f(p^-), f(p^+)]) \leq \max(\pi[p^-, p^+]) + \delta = \pi(p^-) + \delta. \tag{3.26}$$

Since $f(p^-) \leq y \leq y^{(a)} \leq f(p^+)$, Equations (3.25) and (3.26) imply

$$\left\{\tau(t) \,:\, t \in [y^{(a)}, f(p^+)]\right\} \subseteq [\pi(p^-)]_\delta = [\pi(p^+)]_\delta,$$

as claimed. Note that the proof holds both if $s_{j+1} < f(p^+)$ or $s_{j+1} \geq f(p^+)$, which is the distinction we will make in the non-trivial subcases. □

From now on, we will **assume the non-trivial (sub)case**, i.e. $\pi(p^+) > \tau(z_{\min}^{(a+1)}) + \delta$. Our matching scheme is based on a stopping parameter $\overline{a}$, which (intuitively) depends on whether f matched some point on the missing subcurve $\pi[p^-, p^+]$ to a signature vertex $\tau(s_{j+1})$ of τ.

Definition 3.23 (**Stopping parameter $\overline{a}$**). *If $s_{j+1} \geq f(p^+)$, then let $\overline{a}$ be the minimal value of the index a satisfying $f(p^+) \leq y^{(a)}$. Otherwise, let $\overline{a}$ be the minimal value of a such that $y^{(a)} = y^{(a+1)} \leq s_{j+1}$.*

We show that the stopping parameter is well-defined by the following claim.

Claim 3.24. *The stopping parameter $\overline{a}$ (cf. Definition 3.23) is well-defined, and the iterative part of the matching scheme (Equation (3.19)) is valid for $a \leq \overline{a}$.*

Proof. We first argue that there must be a value of a such that $t_{\min}^{(a+1)} = y^{(a)} = y^{(b)}$ for any $b > a$. Recall that by our initial assumptions, we have chosen the signature edge, such that $z_{\min} \in [s_j, s_{j+1}]$, and thus $z_{\min} \leq t_{\min} \leq s_{j+1}$. As a consequence, Claim 3.13 testifies that in the non-trivial case, the point $\tau(y)$ exists and is well-defined. We defined $y^{(1)} = y$ and for $a > 1$ we defined

$$y^{(a)} = \min\{t \in [t_{\min}^{(a)}, s_{j+1}] \colon \tau(t) = \pi(p_{\max}) - \delta\}.$$

Since by Claim 3.13, $\tau(s_{j+1}) \geq \pi(p_{\max}) - \delta = \tau(y^{(a)})$, there must be a value of a, such that

$$\min(\tau[y^{(a)}, s_{j+1}]) \geq \tau(y^{(a)}).$$

Let this value of a be denoted $\widehat{a}$. In this case, it follows by definition that $t_{\min}^{(\widehat{a}+1)} = y^{(\widehat{a})}$. This further implies that $y^{(\widehat{a}+1)} = y^{(\widehat{a})}$ and $t_{\min}^{(\widehat{a}+2)} = y^{(\widehat{a}+1)}, \ldots$. Therefore, by induction it is $y^{(\widehat{a})} = y^{(b)}$, for all $b > \widehat{a}$. Now, if $s_{j+1} < f(p^+)$, then the above analysis implies that the stopping parameter $\overline{a}$ is well-defined.

However, if $s_{j+1} \geq f(p^+)$, we defined $\overline{a}$ to be the minimal value of a such that $f(p^+) \leq y^{(a)}$. Now it might happen that $y^{(\widehat{a})} < f(p^+) \leq s_{j+1}$. In this case, there exists no value of a, such that $f(p^+) \leq y^{(a)}$, thus $y^{(\overline{a})}$ does not exist. We can reduce this case to the trivial case (Case 5(i)) as follows. By Claim 3.13, $\tau(s_{j+1}) \geq \pi(p_{\max}) - \delta$ and by Property 1(iii), $\tau(s_{j+1})$ must be a maximum on $\tau[s_j, s_{j+2}]$. Thus, by definition of $t_{\min}^{(\widehat{a})}$, we would have $\tau\left(z_{\min}^{(\widehat{a})}\right) = \tau\left(t_{\min}^{(\widehat{a})}\right) = \tau(y^{(\widehat{a}-1)}) = \pi(p_{\max}) - \delta \geq \pi(p^-) - \delta$, which is Case 5(i) (the trivial (sub)case).

Therefore, in the non-trivial case, $\overline{a}$ is well-defined. The validity of the iterative part of the matching scheme (Equation (3.19)) for $a \leq \overline{a}$ follows from Claim 3.16 and Claim 3.21, as in the proof of Claim 3.22. □

It remains to complete the matching scheme for the unmatched subcurves $\pi[x^{(\overline{a})}, p^-]$ and $\tau[y^{(\overline{a})}, f(p^+)]$. In order to set up a case analysis with a similar structure as before, we define

$$\begin{aligned} q' &= \min\left\{g(\pi(p_{\max})), g(\tau(t_{\min}^{(\overline{a})}) + \delta)\right\} \\ q'' &= \min\left\{g(\pi(p_{\max})), g(\tau(s_{j+1}) - \delta)\right\} \end{aligned}$$

The exact case distinction is specified in Table 3.1: (i) trivial case (see Claim 3.22), (ii) π stays level, (iii) π tends upwards, (iv) π tends downwards, (v) unmatched signature vertex and π tends upwards, (vi) unmatched signature vertex and π tends downwards. We have

to formally prove that the case analysis is complete, and to prove correctness in each of these subcases.

case	definition	intended matching
5(i)	$\exists a : \pi(p^-) \leq \tau(z_{\min}^{(a+1)}) + \delta$	$\pi[x^{(a)}, p^-] \Leftrightarrow \tau(y^{(a)})$ $\pi(p^+) \Leftrightarrow \tau[y^{(a)}, f(p^+)]$
5(ii)	$p^+ \leq f^{-1}(y^{(\overline{a})}) \leq q'$	$\pi[x^{(\overline{a})}, p^-] \Leftrightarrow \tau(y^{(\overline{a})})$ $\pi[p^+, f^{-1}(y^{(\overline{a})})] \Leftrightarrow \tau(y^{(\overline{a})})$
5(iii)	$p^+ \leq q' < f^{-1}(y^{(\overline{a})})$ and $q' = g(\pi(p_{\max}))$	This case can be reduced to Case 5(i)
5(iv)	$p^+ \leq q' < f^{-1}(y^{(\overline{a})})$ and $q' = g(\tau(t_{\min}^{(\overline{a})}) + \delta)$	$\pi[x^{(\overline{a}-1)}, p^-] \Leftrightarrow \tau(y^{(\overline{a}-1)})$ $\pi[p^+, q'] \Leftrightarrow \tau(y^{(\overline{a}-1)})$ $\pi(q') \Leftrightarrow \tau[y^{(\overline{a}-1)}, f(q')]$
5(v)	$p^+ > f^{-1}(y^{(\overline{a})})$ and $q'' = g(\pi(p_{\max}))$ For the matching scheme, let $x' = \min\{p \in [x^{(\overline{a})}, p_{\max}] : \pi(p) = \tau(s_{j+1}) - \delta\}$ $y' = \max\{t \in [0, f(q'')] : \tau(t) = \tau(y)\}$ $z = \max\{p^+, f^{-1}(y')\}$	$\pi[x^{(\overline{a})}, x'] \Leftrightarrow \tau(y^{(\overline{a})})$ $\pi(x') \Leftrightarrow \tau[y^{(\overline{a})}, y']$ $\pi[x', p^-] \Leftrightarrow \tau(y')$ $\pi[p^+, z] \Leftrightarrow \tau(y')$ $\pi[z] \Leftrightarrow \tau[y', f(p^+)]$
5(vi)	$p^+ > f^{-1}(y^{(\overline{a})})$ and $q'' = g(\tau(s_{j+1}) - \delta)$	$\pi[x^{(\overline{a})}, p^-] \Leftrightarrow \tau(y^{(\overline{a})})$ $\pi[p^+, q''] \Leftrightarrow \tau(y^{(\overline{a})})$ $\pi(q'') \Leftrightarrow \tau[y^{(\overline{a})}, f(q'')]$

Table 3.1: Subcases for Case 5: (i) trivial case (Claim 3.22), (ii) π stays level, (iii) π tends upwards, (iv) π tends downwards, (v) unmatched signature vertex and π tends upwards, (vi) unmatched signature vertex and π tends downwards. Examples of these cases are shown in Figure 3.7 and Figure 3.8.

Claim 3.25. *The case distinction of subcases of Case 5 (cf. Table 3.1) is complete.*

Proof. We assume that we do not have the trivial case (Case 5(i)), i.e. $\pi(p^+) > \tau(z_{\min}^{(a+1)}) + \delta$, for all $a \geq 2$. If $f(p^+) \leq y^{(\overline{a})}$ (also $f(p^+) \leq y^{(\overline{a})} \leq s_{j+1}$), we get one of Case 5(ii)-(iv). Note that if $q' = p_\infty$, then we have Case 5(ii). Thus, this part of the case distinction is complete.

Otherwise we have $f(p^+) > y^{(\overline{a})}$ (also $f(p^+) > s_{j+1} \geq y^{(\overline{a})}$). In this case, we get one of Case 5(v)-(vi). To show the completeness, we need to show that there is no case omitted.

In the following, we argue that, if the subcurve of τ specified by the parameter interval $[f(p^-), f(p^+)]$ contains the signature vertex at s_{j+1}, it must be that

$$\tau(s_{j+1}) - \delta \in \{\pi(p) \ : \ p \in [p^+, 1]\} \tag{3.27}$$

and thus, $q'' \neq p_\infty$. Equation (3.27) implies that $\pi(q'')$ is one of $\{\pi(p_{\max}), \tau(s_{j+1}) - \delta\}$.

From $s_{j+1} \in [f(p^-), f(p^+)]$ it follows that $\pi(p^+) \geq \pi(f^{-1}(s_{j+1}))$. By our initial assumptions it is

$$\max(\pi[p^-, p^+]) = \pi(p^+).$$

Assume that $s_{j+2} \neq 1$, i.e., the next signature vertex after $\tau(s_{j+1})$ is not the last signature vertex. In this case, by Property 1(i) and Property 2, we have

$$\pi(p^+) \geq \pi(f^{-1}(s_{j+1})) \geq \tau(s_{j+1}) - \delta \geq \tau(s_{j+2}) + \delta \geq \pi(f^{-1}(s_{j+2})).$$

Since π is continuous, this implies that there must exist a point $\pi(t)$, with $p^+ \leq t$ and $\pi(t) = \tau(s_{j+1}) - \delta$, i.e. such that Equation (3.27) is satisfied.

Now, assume that $s_{j+2} = 1$. In this case, we have by the theorem statement that $\pi(p_i) \notin [\tau(s_{j+2})]_{4\delta}$. It must be that either $\tau(s_{j+1}) \geq \pi(p_i) > \tau(s_{j+2}) + 4\delta$, or $\pi(p_i) < \tau(s_{j+2}) - 4\delta \leq \tau(s_{j+1}) - 5\delta$. The second case is not possible, since by Claim 3.15 and by Property 2 we have

$$\begin{aligned} \pi(p_i) &\geq \tau(f(p_i)) - \delta \geq \tau(t_{\min}) - \delta \overset{(3.7)}{\geq} \pi(p_{\max}) - 4\delta \\ &\geq \pi(p^+) - 4\delta \geq \pi(f^{-1}(s_{j+1})) - 4\delta \geq \tau(s_{j+1}) - 5\delta. \end{aligned}$$

Thus, $\tau(s_{j+1}) > \tau(s_{j+2}) + 4\delta$, and, as in the case $s_{j+2} \neq 1$, we have that

$$\pi(p^+) \geq \pi(f^{-1}(s_{j+1})) \geq \tau(s_{j+1}) - \delta > \tau(s_{j+2}) + 3\delta \geq \pi(f^{-1}(s_{j+2})) + 2\delta \geq \pi(f^{-1}(s_{j+2})).$$

By the continuity of π, there is a point on π that satisfies Equation (3.27). Therefore, one of Case 5(v)-(vi) occurs, as claimed. □

The cases with $f(p^+) \leq y^{(\overline{a})} \leq s_{j+1}$

We prove now the subcases (ii), (iii), and (iv) of Case 5 (cf. Table 3.1), where the signature vertex $\tau(s_{j+1})$ does not belong to the subcurve of τ, that was matched by f to the missing part of π.

Claim 3.26 (**Correctness of Case 5(ii)**). *Assume $y < f(p^+) \leq y^{(\overline{a})} \leq s_{j+1}$ and $p^+ \leq f^{-1}(y^{(\overline{a})}) \leq q'$ (Case 5(ii)). Then the matching scheme given by Equation (3.28) is valid.*

Proof. We want to use the matching

$$\begin{cases} \pi[x^{(\overline{a})}, p^-] & \Leftrightarrow \quad \tau(y^{(\overline{a})}), \\ \pi[p^+, f^{-1}(y^{(\overline{a})})] & \Leftrightarrow \quad \tau(y^{(\overline{a})}), \end{cases} \tag{3.28}$$

and for the suffix curves $\pi[f^{-1}(y^{(\overline{a})}), 1]$ and $\tau[y^{(\overline{a})}, 1]$ we reuse f, since $y^{(\overline{a})} \geq f(p^+)$. By Claim 3.16, the Fréchet distance between $\pi[x^{(\overline{a})}, p^-]$ and $\tau(y^{(\overline{a})})$ is at most δ, implying the validity of the first row of (3.28).

Since $t_{\min}^{(\overline{a}-1)}$ always exists, due to $y^{(1)} \leq s_{j+1}$ by Claim 3.14, we have by Claim 3.15 and the definition of $t_{\min}^{(\overline{a})}$, that

$$\tau(t_{\min}^{(\overline{a})}) + \delta \geq \tau(t_{\min}^{(\overline{a}-1)}) + \delta \geq \tau(t_{\min}) + \delta \geq \pi(x) \geq \pi(p_{\max}) - 2\delta.$$

By our case distinction, it is $p^+ \leq f^{-1}(y^{(\overline{a})}) \leq q'$, and thus

$$\begin{aligned} \left\{\pi(p) \,:\, p \in [p^+, f^{-1}(y^{(\overline{a})})]\right\} &\subseteq [\tau(t_{\min}^{(\overline{a})}) + \delta, \pi(p_{\max})] \subseteq [\tau(t_{\min}^{(\overline{a}-1)}) + \delta, \pi(p_{\max})] \\ &\subseteq [\pi(p_{\max}) - 2\delta, \pi(p_{\max})] = [\tau(y^{(\overline{a})})]_\delta. \end{aligned}$$

This implies that also the second matching of (3.28) is valid, as claimed. □

Claim 3.27 (**Correctness of Case 5(iii)**). *Assume $y < f(p^+) \leq y^{(\overline{a})} \leq s_{j+1}$, and let in this case $p^+ \leq q' < f^{-1}(y^{(\overline{a})})$ and $q' = g(\pi(p_{\max}))$ (Case 5(iii)). Then the conditions of Case 5(i) are satisfied.*

Proof. By the case definition it is $f(p^+) \leq f(q') < y^{(\overline{a})}$ and $q' = g(\pi(p_{\max}))$. We can reduce this case to Case 5(i) (the trivial case) as follows.

Let b be the maximal value of a such that $f(q') \in [y^{(a)}, y^{(\overline{a})}]$. By Property 2 it must be that $\tau(f(q')) \geq \pi(q') - \delta = \pi(p_{\max}) - \delta = \tau(y^{(b)})$. By the definition of $y^{(a)}$, for any a, τ goes upwards in $\tau(y^{(a)})$, then intersects $\pi(p_{\max}) - \delta$ downwards, and goes upwards again in $\tau(y^{(a+1)})$. Thus, by the choice of b, it is $\min(\tau[y^{(b)}, f(q')]) \geq \pi(p_{\max}) - \delta$.

By our case distinction and the choice of b, $f(p^+) \in [y^{(b)}, f(q')]$. Thus,

$$\tau(z_{\min}^{(b+1)}) = \min(\tau[y^{(b)}, f(p^+)]) \geq \min(\tau[y^{(b)}, f(q')]) \geq \pi(p_{\max}) - \delta \geq \pi(p^-) - \delta,$$

and we are again in Case 5(i). □

Claim 3.28 (**Correctness of Case 5(iv)**). *Assume $y < f(p^+) \leq y^{(\overline{a})} \leq s_{j+1}$, and let in this case $p^+ \leq q' < f^{-1}(y^{(\overline{a})})$ and $q' = g(\tau(t_{\min}^{(\overline{a})}) + \delta)$ (Case 5(iv)). Then the matching scheme given by Equation (3.29) is valid.*

Proof. By our case distinction, it is $p^+ \leq q' < f^{-1}(y^{(\overline{a})})$ and $q' = g(\tau(t_{\min}^{(\overline{a})}) + \delta)$. In this case, we rollback the last two matchings of the iterative part of the matching scheme (Equation (3.19)), and instead end with $a = \overline{a} - 1$. Thus, we are left with the unmatched subcurves $\pi[x^{(\overline{a}-1)}, p^-]$ and $\tau[y^{(\overline{a}-1)}, f(p^+)]$. We intend to use the following matching:

$$\begin{cases} \pi[x^{(\overline{a}-1)}, p^-] & \Leftrightarrow & \tau(y^{(\overline{a}-1)}), \\ \pi[p^+, q'] & \Leftrightarrow & \tau(y^{(\overline{a}-1)}), \\ \pi(q') & \Leftrightarrow & \tau[y^{(\overline{a}-1)}, f(q')], \end{cases} \tag{3.29}$$

and to complete the matching scheme with the suffix curves $\pi[q', 1]$ and $\tau[f(q'), 1]$, reusing the matching f, since $p^+ \leq q'$.

The validity of the first matching in (3.29) follows directly from Claim 3.16, since $x^{(\overline{a}-1)} > x$. By the definition of $t_{\min}^{(\overline{a})}$ and Property 1(iv), it has to hold

$$\tau(t_{\min}^{(\overline{a})}) + 2\delta \geq \tau(y^{(\overline{a}-1)}) = \pi(p_{\max}) - \delta. \tag{3.30}$$

Then, by the definition of q' and our case distinction, it is

$$\left\{\pi(p) \,:\, p \in [p^+, q']\right\} \subseteq [\tau(t_{\min}^{(\overline{a})}) + \delta, \pi(p_{\max})] \overset{(3.30)}{\subseteq} [\pi(p_{\max}) - \delta]_\delta = [\tau(y^{(\overline{a}-1)})]_\delta.$$

This proves the validity of the second matching in (3.29). Finally, Claim 3.21 implies the validity of the third matching in (3.29), since $[y^{(\overline{a}-1)}, f(q')] \subseteq [y^{(\overline{a}-1)}, y^{(\overline{a})}]$, and $\pi(q') = \tau(t_{\min}^{(\overline{a})}) + \delta$. This closes the proof of Case 5(iv). □

We have now analyzed Case 5(i)-(iv), and showed by Claim 3.22, Claim 3.26, Claim 3.27, and Claim 3.28 that, in these cases, Lemma 3.7 is correct. Examples of these cases are shown in Figure 3.7. We now move on to prove correctness of the remaining cases Case 5(v) and Case 5(vi), where the signature vertex $\tau(s_{j+1})$ was in the part of the matching f that needs to be repaired.

The cases with $f(p^+) > s_{j+1} \geq y^{(\overline{a})}$

Claim 3.29 (**Correctness of Case 5(v)**). *Let $f(p^+) > s_{j+1} \geq y^{(\overline{a})} \geq y$ and assume $q'' = g(\pi(p_{\max}))$ (Case 5(v)). Then the matching scheme given by Equation (3.31) is valid.*

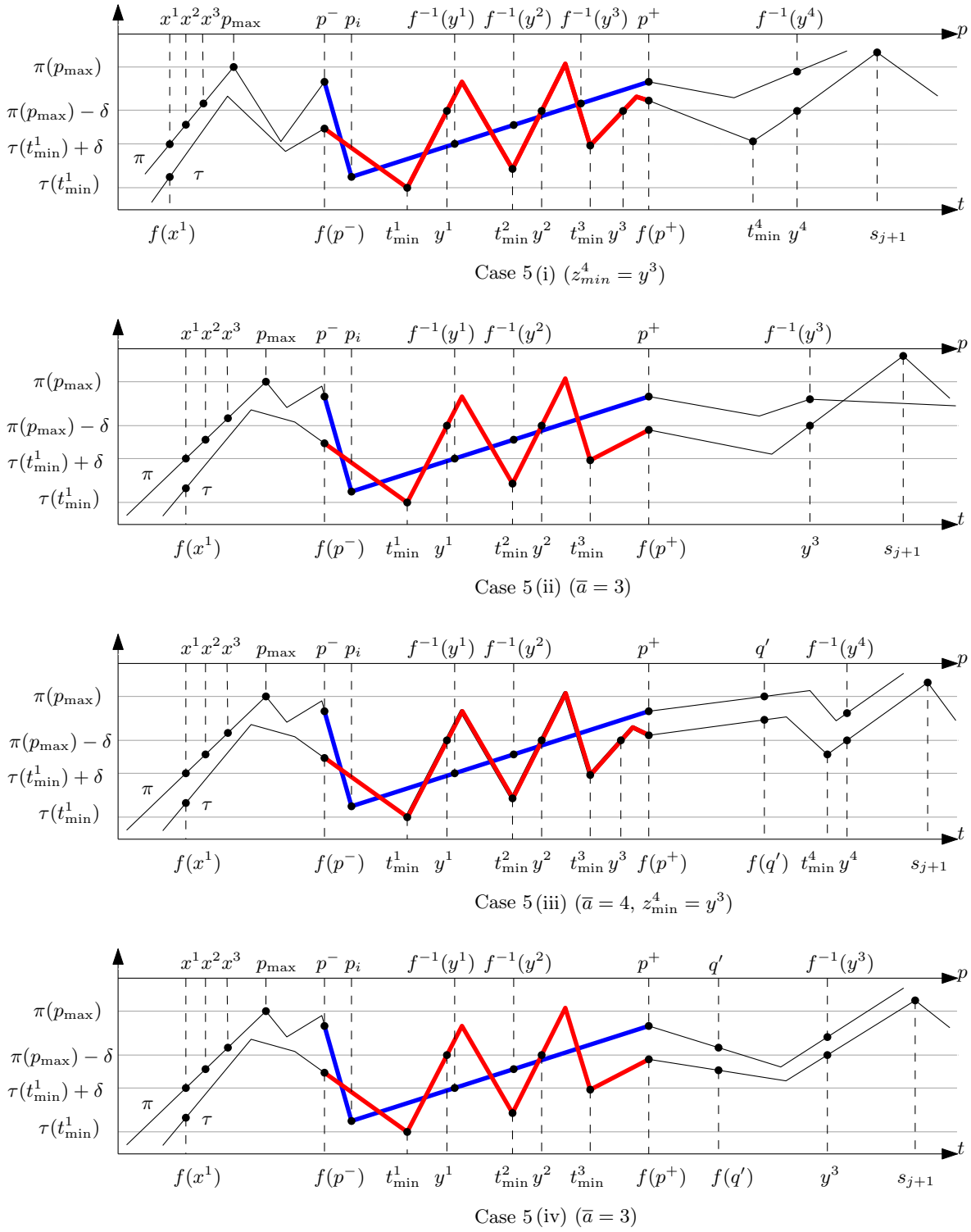
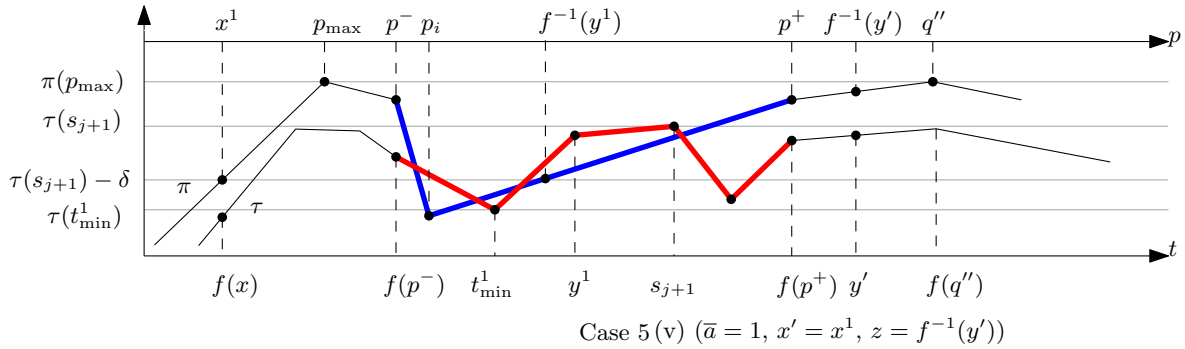
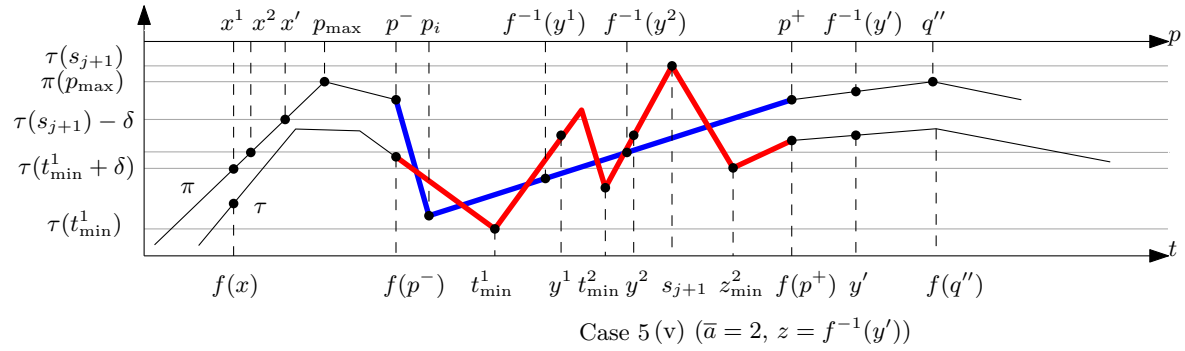


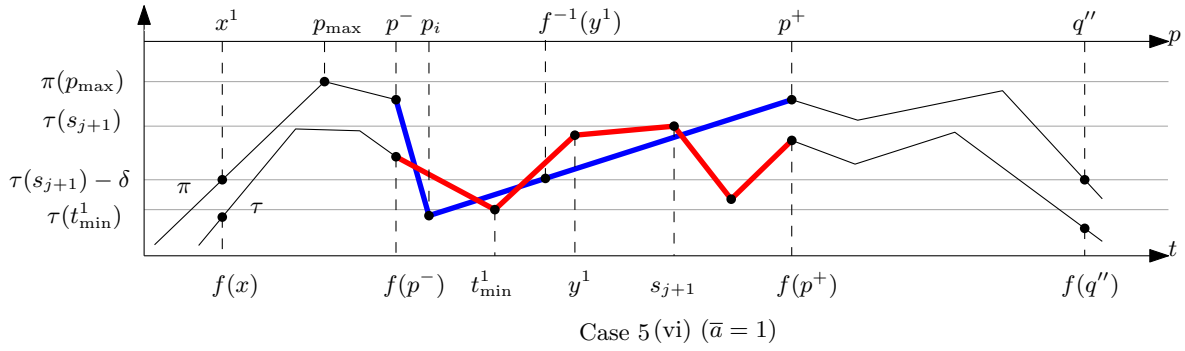
Figure 3.7: Examples of Case 5(i)-(iv). The broken part of the matching f is indicated by thick lines. The indices in the exponents are written without brackets for simplicity.



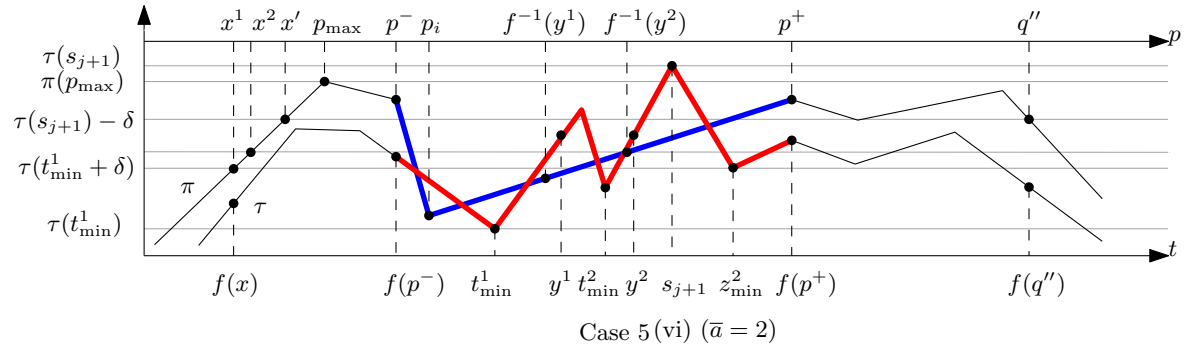
Case 5 (v) ($\overline{a} = 1$, $x' = x^1$, $z = f^{-1}(y')$)



Case 5 (v) ($\overline{a} = 2$, $z = f^{-1}(y')$)



Case 5 (vi) ($\overline{a} = 1$)



Case 5 (vi) ($\overline{a} = 2$)

Figure 3.8: Examples of Case 5(v)-(vi). The broken part of the matching f is indicated by thick lines. The indices in the exponents are written without brackets for simplicity.

Proof. By our case distinction, it holds that $f(p^+) > y^{(\overline{a})}$ and $q'' = g(\pi(p_{\max}))$. We introduce the following additional notation:

$$\begin{aligned} x' &= \min\{p \in [x^{(\overline{a})}, p_{\max}] \colon \pi(p) = \tau(s_{j+1}) - \delta\}, \\ y' &= \max\{t \in [0, f(q'')] \colon \tau(t) = \tau(y)\}, \\ z &= \max\{p^+, f^{-1}(y')\}. \end{aligned}$$

We intend to use the following matching scheme

$$\begin{cases} \pi[x^{(\overline{a})}, x'] & \Leftrightarrow \quad \tau(y^{(\overline{a})}) \\ \pi(x') & \Leftrightarrow \quad \tau[y^{(\overline{a})}, y'] \\ \pi[x', p^-] & \Leftrightarrow \quad \tau(y') \\ \pi[p^+, z] & \Leftrightarrow \quad \tau(y') \\ \pi[z] & \Leftrightarrow \quad \tau[y', f(p^+)] \end{cases} \tag{3.31}$$

Observe that in this case $q'' = q = g(\pi(p_{\max}))$, as in Case 3 (cf. Equation (3.13)). Therefore, y' and z are the same as in Case 3 and they must exist (cf. Equation (3.15)). We argue that x' must also exist. Recall that by our case distinction $f(p^-) \leq s_{j+1} \leq f(p^+)$. By Property 1(iii) and Property 2, it follows that

$$\tau(s_{j+1}) - \delta = \max(\tau[f(p^-), f(p^+)]) - \delta \leq \max(\pi[p^-, p^+]) \leq \pi(p_{\max}). \tag{3.32}$$

By the definition of $x^{(a)}$ it is $x^{(a)} \leq \tau(s_{j+1}) - \delta$, for all $a \geq 2$. Thus, $\pi(x^{(\overline{a})}) \leq \tau(s_{j+1}) - \delta \overset{(3.32)}{\leq} \pi(p_{\max})$, and by the continuity of π, x' has to exist.

Now we need to prove the validity of the matching scheme (3.31). The first line follows from Claim 3.16, since $x^{(\overline{a})} \geq x$. For the second line in (3.31), we need to prove that

$$\left\{\tau(t) \ : \ t \in [y^{(\overline{a})}, y']\right\} \subseteq [\tau(s_{j+1}) - 2\delta, \tau(s_{j+1})] = [\pi(x')]_\delta.$$

The upper bound follows from Property 1(iii), thus, $\max(\tau[y^{(\overline{a})}, y']) \leq \tau(s_{j+1})$. As for the lower bound, by the definition of the stopping parameter (when $f(p^+) > s_{j+1}$) and using Equation (3.32), we have

$$\min(\tau[y^{(\overline{a})}, s_{j+1}]) = \tau(y^{(\overline{a})}) = \pi(p_{\max}) - \delta \overset{(3.32)}{\geq} \tau(s_{j+1}) - 2\delta. \tag{3.33}$$

By Claim 3.10,

$$\min(\tau[s_{j+1}, f(p^+)]) \geq \tau(s_{j+1}) - 2\delta. \tag{3.34}$$

By Property 2 and by our case distinction ($\tau(y')$ is the last point before $\tau(q'')$ with the specified value), we have

$$\min(\tau[f(p^+), y')]) = \min(\tau[f(p^+), f(q'')]) \geq \min(\pi[p^+, q'']) - \delta \geq \tau(s_{j+1}) - 2\delta. \quad (3.35)$$

Therefore, by Equations (3.33), (3.34), and (3.35), $\min(\tau[y^{(\overline{a})}, y']) \geq \tau(s_{j+1}) - 2\delta$, and the second matching in (3.31) is valid as well. The validity of the third matching is implied by Claim 3.16, since $x' > x$. For the last two matchings we can apply the respective part of the proof of Case 3 (for matching presented in Equation (3.16), cf. Claim 3.19) verbatim. □

Claim 3.30 (Correctness of Case 5(vi)). *Let $f(p^+) > s_{j+1} \geq y^{(\overline{a})} \geq y$ and assume $q'' = g(\tau(s_{j+1}) - \delta)$ (Case 5(vi)). Then the matching scheme given by Equation (3.36) is valid.*

Proof. In this case it is $f(p^+) > y^{(\overline{a})}$ and $q'' = g(\tau(s_{j+1}) - \delta)$. We will use the following matching scheme:

$$\begin{cases} \pi[x^{(\overline{a})}, p^-] & \Leftrightarrow & \tau(y^{(\overline{a})}) \\ \pi[p^+, q''] & \Leftrightarrow & \tau(y^{(\overline{a})}) \\ \pi(q'') & \Leftrightarrow & \tau[y^{(\overline{a})}, f(q'')] \end{cases} \quad (3.36)$$

The validity of the first matching in (3.36) follows from Claim 3.16, since $x^{(\overline{a})} \geq x$. By Property 1(iii) and Property 2 it is $\tau(s_{j+1}) \geq \pi(p_{\max}) - \delta$. Then, by our case distinction,

$$\{\pi(p) \ : \ p \in [p^+, q'']\} \subseteq [\tau(s_{j+1}) - \delta, \pi(p_{\max})] \subseteq [\tau(y^{(\overline{a})}]_\delta).$$

Thus, the second matching in (3.36) is valid as well. For the third matching in (3.36) we need to prove that

$$\left\{\tau(t) \ : \ t \in [y^{(\overline{a})}, f(q'')]\right\} \subseteq [\tau(s_{j+1}) - 2\delta, \tau(s_{j+1})] = [\pi(q'')]_\delta. \quad (3.37)$$

Again, as in Case 5(v), from Equations (3.33) and (3.34), it follows that

$$\left\{\tau(t) \ : \ t \in [y^{(\overline{a})}, f(p^+)]\right\} \subseteq [\tau(s_{j+1}) - 2\delta, \tau(s_{j+1})]. \quad (3.38)$$

By Property 2 and by our case distinction

$$\min(\tau[f(p^+), f(q'')]) \geq \min(\pi[p^+, q'']) - \delta \geq \tau(s_{j+1}) - 2\delta \quad (3.39)$$

Equation (3.39) also implies that $f(q'') < s_{j+2}$, by Property 1(i). Thus, by Property 1(iii), we conclude

$$\max(\tau[f(p^+), f(q'')]) \leq \tau(s_{j+1}). \tag{3.40}$$

Equations (3.38), (3.39), and (3.40) together imply the correctness of (3.37), and thus, the validity of the last matching in (3.36). □

We now proved correctness of the last two cases (Case 5(v) and Case 5(vi)), and showed by Claim 3.29, and Claim 3.30 that, in these cases, Lemma 3.7 is correct. Examples of these cases are shown in Figure 3.8.

3.3.4 Boundary cases

It remains to prove the boundary cases, which we have ruled out so far by Assumption 3. There are three boundary cases:
(B1) $s_j = 0$ and $s_{j+1} = 1$ (there is only one signature edge),
(B2) $s_j = 0$ and $s_{j+1} < 1$ (the first signature edge),
(B3) $s_j > 0$ and $s_{j+1} = 1$ (the last signature edge).

To prove the claim in each of these cases, we can use the proof we have done under Assumption 3 verbatim, with minor modifications. Throughout the proof, we used s_j only in its function as the minimum on the signature edge $\overline{s_j s_{j+1}}$, and s_{j+1} only in its function as the maximum on the signature edge, respectively. Thus, let

$$s_{\min} = \operatorname*{arg\,min}_{s \in [s_j, s_{j+1}]} \tau(s) \quad \text{and} \quad s_{\max} = \operatorname*{arg\,max}_{s \in [s_j, s_{j+1}]} \tau(s).$$

Assumption 1 and Assumption 2 remain valid. The next claim relates the point $\tau(f(p_i))$, that was matched to the removed vertex on π, to $\tau(s_{\min})$ and $\tau(s_{\max})$.

Claim 3.31. *In each of the cases* (B1), (B2), *and* (B3), *it holds for the removed point* $\pi(p_i)$ *that* $f(p_i) \in [s_{\min}, s_{\max}]$ *and* $\tau(s_{\max}) - \tau(s_{\min}) \geq 4\delta$.

Proof. By the theorem statement and by Definition 3.3, it holds that

$$\pi(p_i) \notin [v_1]_{4\delta} \cup [v_\ell]_{4\delta} = [\tau(0)]_{4\delta} \cup [\tau(1)]_{4\delta}, \tag{3.41}$$

i.e., the removed vertex $\pi(p_i)$ lies very far from the endpoints of the curve τ. At the same time, by Definition 3.3, in case $s_j = 0$ ((B1) and (B2)),

$$\tau(0) \geq \tau(s_{\min}) \geq \tau(0) - \delta, \tag{3.42}$$

and, in case $s_{j+1} = 1$ ((B1) and (B3)),

$$\tau(1) \leq \tau(s_{\max}) \leq \tau(1) + \delta. \tag{3.43}$$

For the sake of a contradiction, assume that $f(p(i)) < s_{\min}$. In the case when $s_j \neq 0$ (B3), by Assumption 1 it is $s_{\min} = s_j$ (a contradiction). In other two cases, by the direction-preserving property of Definition 3.3 and by Property 2, it is

$$\tau(0) - 2\delta \leq \tau(f(p_i)) - \delta \leq \pi(p_i) \leq \tau(f(p_i)) + \delta \leq \tau(0) + 2\delta,$$

a contradiction to Equation (3.41). Analogously, we conclude that it cannot be $f(p_i) > s_{\max}$, and thus, it holds that $f(p_i) \in [s_{\min}, s_{\max}]$, as claimed.

By the direction-preserving property of Definition 3.3, we conclude further, making distinction on relation between $\pi(p_i)$, and $\tau(0)$ and $\tau(1)$, that:

(i) If $\tau(0) + 4\delta < \tau(f(p_i))$ and $\tau(1) + 4\delta < \tau(f(p_i))$, then we have in the cases (B1) and (B2), that

$$\tau(s_{\min}) + 4\delta \overset{(3.42)}{\leq} \tau(0) + 4\delta < \tau(f(p_i)) \leq \tau(s_{\max}), \tag{3.44}$$

and in the case (B3), that

$$\tau(s_{\min}) + 4\delta \overset{(3.43)}{\leq} \tau(1) + 4\delta < \tau(f(p_i)) \leq \tau(s_{\max}). \tag{3.45}$$

Equations (3.44) and (3.45) imply $\tau(s_{\max}) - \tau(s_{\min}) > 4\delta$.

(ii) If $\tau(0) + 4\delta > \tau(f(p_i))$ and $\tau(1) - 4\delta > \tau(f(p_i))$, then in the cases (B1) and (B2) Equation (3.44) remains valid. In the case (B3) we have

$$\tau(s_{\min}) \leq \tau(f(p_i)) < \tau(1) - 4\delta \overset{(3.43)}{\leq} \tau(s_{\max}) - 4\delta. \tag{3.46}$$

Equations (3.44) and (3.46) imply $\tau(s_{\max}) - \tau(s_{\min}) > 4\delta$.

The remaining two cases: $\tau(0) - 4\delta > \tau(f(p_i))$ and $\tau(1) - 4\delta > \tau(f(p_i))$, and $\tau(0) + 4\delta > \tau(f(p_i))$ and $\tau(1) - 4\delta > \tau(f(p_i))$, respectively, are equivalent to the two cases we analyzed above. Thus, the second part of the claim is proved. □

We replace Property 1 with the following property.

Property 3 (**Signature (boundary case)**)**.**
(i) $\tau(s_{\max}) - \tau(s_{\min}) > 2\delta$,

(ii) $\tau(s_{\min}) = \min(\tau[s_{j-1}, s_{j+1}])$ *(if* $s_j = 0$*, then* $\tau(s_{\min}) = \min(\tau[0, s_{j+1}])$*),*
(iii) $\tau(s_{\max}) = \max(\tau[s_j, s_{j+2}])$ *(if* $s_{j+1} = 1$*, then* $\tau(s_{\max}) = \max(\tau[s_j, 1])$*),*
(iv) $\tau(t') - 2\delta \leq \tau(t'')$ *for* $s_{\min} \leq t' < t'' \leq s_{\max}$*,*
(v) if $s_j = 0$*, then* $\tau(s_{\max}) - \tau(s_{j+2}) > 2\delta$*.*

Property 3(ii), (iii), (iv), and (v) hold by Definition 3.3. Property 3(i) follows from Claim 3.31.

Instead of Claim 3.10 we use the claim

Claim 3.32. *If* $s_{\max} \in [f(p^-), f(p^+)]$*, then*

$$\{\tau(t) \ : \ t \in [s_{\max}, f(p^+)]\} \subseteq [\tau(s_{\max}) - 2\delta, \tau(s_{\max})].$$

Instead of Claim 3.11 we use the claim

Claim 3.33. *It holds that* $s_{\min} \notin [f(p^-), f(p^+)]$*.*

The correctness of Claim 3.32 and Claim 3.33 follows by taking the proofs of Claim 3.10 and Claim 3.11, respectively, and by replacing s_j with $s_{\min}$, and replacing s_{j+1} with $s_{\max}$. This is enabled by Claim 3.31. The correctness of Lemma 3.7 follows in the boundary cases (B1), (B2) and (B3), by replacing s_j with $s_{\min}$, and replacing s_{j+1} with $s_{\max}$.

Therefore, we have proved the correctness of Lemma 3.7 in all possible cases, as well as that our case analysis is complete.

3.4 On computing signatures

In this section we discuss how to compute signatures efficiently. Our signatures have a unique hierarchical structure as testified by Lemma 3.34. Together with the concept of vertex permutations (Definition 3.35), this allows us to construct a data structure, which supports efficient queries for the signature of a given size (Theorem 3.39). If the parameter δ is given, we can compute a signature in linear time using Algorithm 1. Furthermore, we show that our signatures are approximate simplifications in Lemma 3.41.

3.4.1 Computing signatures of a given size

We consider first the case of computing the signatures when a size of signature is given.

Lemma 3.34. *Given a polygonal curve* $\tau : [0, 1] \to \mathbb{R}$ *with vertices in general position, there exists a series of signatures* $\sigma_1, \sigma_2, \ldots, \sigma_k$ *and corresponding parameters* $0 = \delta_1 < \delta_2 < \cdots < \delta_{k+1}$*, such that*

(i) σ_i *is a* δ*-signature of* τ *for any* $\delta \in [\delta_i, \delta_{i+1})$,
(ii) the vertex set of σ_{i+1} *is a subset of the vertex set of* σ_i,
(iii) σ_k *is the linear interpolation of* $\tau(0)$ *and* $\tau(1)$.

Proof. We construct the desired series of signatures by a series of edge contractions. Given a curve $\sigma = w_1, \ldots, w_\ell$, the contraction of an edge $\overline{w_j w_{j+1}}$ of σ yields a curve σ', such that:

(a) if $j \in \{2, \ldots, \ell - 2\}$, then the vertices w_{j-1} and w_{j+2} are connected by an edge in σ'; the vertices w_j and w_{j+1} are deleted; the rest of the curve σ' equals σ.

(b) if $j \in \{1, \ell - 1\}$, then the vertices w_1 and w_3 (respectively, $w_{\ell-2}$ and w_ℓ) are connected by an edge in σ'; the vertex w_2 (respectively $w_{\ell-1}$) is deleted; the rest of the curve σ' equals σ.

We start with $\sigma_1 = \tau$, which is clearly a minimal δ_1-signature for $\delta_1 = 0$. We now conceptually increase the signature parameter δ until a smaller signature is possible. In general, let $0 = t_0 < t_1 < \cdots < t_\ell = 1$ be the series of parameters that defines σ_i. Let for $i > 1$ be:

$$\delta_{i+1} = \min \left\{ |\tau(t_1) - \tau(t_2)|, |\tau(t_{\ell-1}) - \tau(t_\ell)|, \min_{2 \le j \le \ell-2} \frac{|\tau(t_j) - \tau(t_{j+1})|}{2} \right\}. \tag{3.47}$$

We contract the edge where the minimum is attained to obtain σ_{i+1}. By the general position assumption, this edge is unique. See Figure 3.9 for an example.

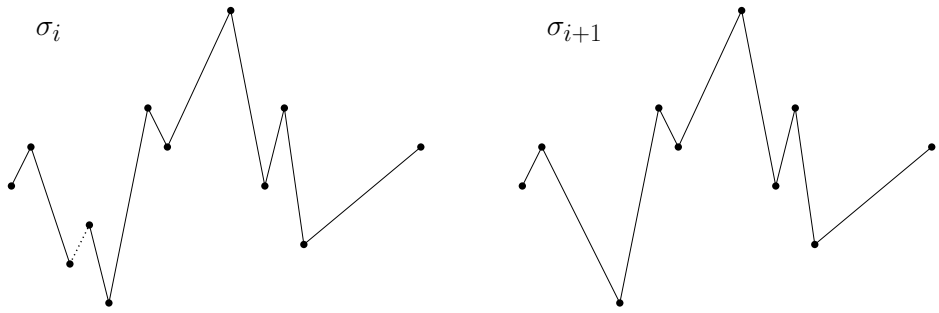


Figure 3.9: Edge contraction from Lemma 3.34. The contracted edge is marked dotted.

We show by induction over i that σ_i is a δ_i-signature. The claim for $i = 1$ holds by definition. We assume that for some i the induction hypothesis holds, and prove that it remains valid for σ_{i+1} and δ_{i+1}. We distinct two cases: whether the contracted edge of σ_i to obtain σ_{i+1} is an internal edge, or is connected to an endpoint.

Case 1: $\overline{\tau(t_j)\tau(t_{j+1})}$ is the contracted edge, with $2 \le j \le \ell - 2$. Observe that

$$\tau(t_j), \tau(t_{j+1}) \in \langle\!\langle \tau(t_{j-1}), \tau(t_{j+2}) \rangle\!\rangle. \tag{3.48}$$

If Equation (3.48) would not hold, say $\tau(t_j) > \tau(t_{j+2}) > \tau(t_{j-1})$ (i.e. the case that $\tau(t_j)$ is a local maximum, the other three cases are analogous), then the contracted edge $\overline{\tau(t_j)\tau(t_{j+1})}$ would not minimize the expression in (3.47). We prove that σ_{i+1} satisfy the conditions of Definition 3.3.

(i) (non-degeneracy): Assume for the sake of contradiction that this property does not hold, i.e., either that $\tau(t_{j-1}) \in \langle\!\langle \tau(t_{j-2}), \tau(t_{j+2}) \rangle\!\rangle$, or that $\tau(t_{j+2}) \in \langle\!\langle \tau(t_{j-1}), \tau(t_{j+3}) \rangle\!\rangle$. We assume the first case. Then, Equation (3.48) would imply that $\tau(t_{j-1}) \in \langle\!\langle \tau(t_{j-2}), \tau(j) \rangle\!\rangle$, which contradicts the non-degeneracy property of σ_i. The second case is symmetric to the first, and follows by analogy.

(ii) (direction-preserving): Since $\delta_i < \delta_{i+1}$, we have to prove this property only for the newly established edge $\overline{\tau(t_{j-1})\tau(t_{j+2})}$ of σ_{i+1} (for the other edges it is inherited from σ_i). The contracted edge was the shortest interior edge of σ_i and by construction we have that

$$|\tau(t_j) - \tau(t_{j+1})| = 2\delta_{i+1} \tag{3.49}$$

For any $s, s' \in [t_j, t_{j+1}]$, $s \leq s'$, it holds that $|\tau(s) - \tau(s')| \leq 2\delta_{i+1}$. Indeed, by induction, the range condition held true for the contracted edge (of σ_i), and by Equation (3.49) its length was $2\delta_{i+1}$. For any $s, s' \in [t_{j-1}, t_j]$, $s \leq s'$, the direction-preserving property of σ_{i+1} holds by induction, and the same holds for the case $s, s' \in [t_{j+1}, t_{j+2}]$. The remaining case is $s, s' \in [t_{j-1}, t_{j+2}]$, $s \leq s'$, where the points $\tau(s)$ and $\tau(s')$ belonged to different edges of σ_i. In this case, the direction-preserving property for σ_{i+1} follows by the range property of σ_i and by Equation (3.49).

(iii) (minimum-edge-length): Equation (3.49) and the choice of the contracted edge imply the minimum-edge-length property for σ_{i+1}.

(iv) (range): Since by induction, the range property was satisfied for σ_i, by the statement in (3.48) the range property cannot be violated by the edge contraction, thus it holds for σ_{i+1} as well.

Case 2: $\overline{\tau(t_j)\tau(t_{j+1})}$ is the contracted edge, with $j = 1$ (the case $j = \ell - 1$ is analogous). For simplicity, let the first three vertices of the current signature σ_i have indices 1, 2, and 3. Again, we prove the conditions of Definition 3.3.

(i) (non-degeneracy): The non-degeneracy property on the vertex $\tau(t_3)$ is not affected by the edge contraction, since $\tau(t_3)$ stays a minimum (resp. maximum) in σ_{i+1} if it was a minimum (resp. maximum) in σ_i. Otherwise, the contracted

edge would not minimize the expression in Equation (3.47). The other vertices of σ_i (except the deleted $\tau(t_2)$) are not affected by the edge contraction.

(ii) (direction-preserving): The arguments for the direction-preserving property are the same as in the case when $j > 1$.

(iii) (minimum-edge-length): $|\tau(t_2) - \tau(t_3)| > 2\delta_{i+1}$ and $|\tau(t_1) - \tau(t_2)| = \delta_{i+1}$ imply the minimum-edge-length property for σ_{i+1}, i.e. that $|\tau(t_1) - \tau(t_3)| > \delta$.

(iv) (range): By induction, the range property and the non-degeneracy property are satisfied for the first two edges of σ_i. Since it holds for the length of the second edge that $|\tau(t_2) - \tau(t_3)| > 2\delta_{i+1}$, it must be that the union of the intervals $[\tau(t_1) - \delta_{i+1}, \tau(t_1) + \delta_{i+1}]$ and $\langle\langle \tau(t_1), \tau(t_3) \rangle\rangle$ spans the range of values on $\tau[t_1, t_3]$. Thus, the range property is satisfied for σ_{i+1}.

By construction it is clear that the vertex set of the σ_{i+1} is a subset of the vertex set of σ_i for each i, as well as that σ_k is the linear interpolation of $\tau(0)$ and $\tau(1)$. This completes the proof of the lemma. □

We call the signatures $\sigma_1, \ldots, \sigma_k$ of Lemma 3.34 the **canonical signatures** of the curve τ. We next define the canonical vertex permutation, following the concept of the vertex permutations of Driemel and Har-Peled [65].

Definition 3.35 (**Canonical vertex permutation**). *Given a curve $\tau : [0, 1] \to \mathbb{R}$ with m vertices in general position, consider its canonical signatures $\sigma_1, \ldots, \sigma_k$ of Lemma 3.34. We call a permutation of the vertices of τ canonical if for any two vertices x, y of τ it holds that if $x \notin \mathcal{V}(\sigma_i)$ (the vertex set of σ_i) and $y \in \mathcal{V}(\sigma_i)$, for some i, then x appears before y in the permutation. Furthermore, we require that the permutation contains a token separator for every σ_i, for $1 \leq i \leq k$, such that σ_i consists of all vertices appearing after the separator.*

A canonical vertex permutation can be computed efficiently. The idea is to simulate the series of edge contractions done in the proof of Lemma 3.34. This is claimed by Lemma 3.36.

Lemma 3.36. *Given a curve $\tau : [0, 1] \to \mathbb{R}$ with m vertices in general position, a canonical vertex permutation can be computed in $O(m \log m)$ time and using $O(m)$ space.*

Proof. Let $w_1, \ldots, w_m$ be the vertices of the curve τ. We build a min-heap from the vertices $w_2, \ldots, w_{m-1}$ using certain keys, that will be defined briefly. We iteratively extract the (one or more) vertices with the current minimum key from the heap and update the keys of their neighboring vertices along the current signature curves. The extracted vertices are recorded in a list L (which is initially empty) in the order of their extraction and will form

the canonical vertex permutation in the end. Before every iteration we append a token separator to L. In this way, all vertices extracted during one iteration are placed between two token separators in L. After the last iteration we again append a token separator and, at last, the two vertices w_1 and w_m.

More precisely, let $v_1, \ldots, v_k$ (a subset of $\{w_2, \ldots, w_{m-1}\}$, renamed for simplicity of notation) denote the vertices contained in the heap at the beginning of one particular iteration, sorted in the order of their appearance along the curve σ. We call the curve

$$\sigma = w_1, v_1, \ldots, v_k, w_m,$$

the *current signature*. For every vertex we keep pointers to the heap elements that represent its current predecessor and successor vertices along the current signature. We also keep these pointers to the virtual elements w_1 and w_m, which are not included in the heap. We define the key $W(v_i)$ for every vertex v_i in the heap as follows:

(i) if $i = 1$, then $W(v_i) = \min\Big(|w_1 - v_1|, \frac{|v_1 - v_2|}{2}\Big)$,

(ii) if $i = k$, then $W(v_i) = \min\Big(|w_m - v_k|, \frac{|v_k - v_{k-1}|}{2}\Big)$, otherwise

(iii) $W(v_i) = \min\Big(\frac{|v_i - v_{i-1}|}{2}, \frac{|v_i - v_{i+1}|}{2}\Big)$.

Initially, the current signature equals τ and initializing these keys takes $O(m)$ time in total. Building a min-heap takes $O(m \log m)$ time. Following the argument from Lemma 3.34, we need to contract the edge(s) with minimum length (where exceptions hold for the first and the last edge). This is captured by the choice of the keys' values above.

We first assume that the minimum is attained for exactly one edge with two endpoints v_i and v_{i+1} in the heap, for some i. In this case, v_i and v_{i+1} are the next two elements to be extracted from the heap and their keys must be equal to $\frac{|v_i - v_{i+1}|}{2}$. Using the pointers to v_{i-1} (unless $i = 1$) and v_{i+2} (unless $i = k$), we now update the key values of these neighbors and update the pointers such that v_{i-1} (respectively, w_1) becomes predecessor to v_{i+2}, and v_{i+2} (respectively, w_m) becomes successor to v_{i-1}. Computing the new key value of one of these neighboring vertices can be done in $O(1)$ time. Updating the keys in the heap takes $O(\log m)$ time per vertex. We can charge every update of the keys and the pointers to the extraction of a neighboring vertex, thus the extraction costs $O(\log m)$ per vertex. Since every vertex is extracted exactly once, we need $O(m \log m)$ time in total. The space $O(m)$ requirement follows from the construction of the min-heap.

If only one vertex v_1 or v_k is extracted during the iteration, which corresponds to the case that an edge adjacent to w_1 or w_m is being contracted, the presented upper bounds remain valid. By the general posititon assumption, it is not possible to have more than one edge (and two vertices) to be extracted simultaneously. If this assumption would be

neglected, then multiple edges of the same length would be contracted all at once. In this case more than two vertices need to be extracted, but the (amortized) cost of extraction per vertex (in particular, the updating of the pointers) would remain $O(\log m)$, as every vertex is extracted exactly once. □

We have computed the canonical vertex permutation by Lemma 3.36. The running time of Lemma 3.36 is optimal, as it is showed in Lemma 3.37.

Lemma 3.37. *Given a curve $\tau : [0,1] \to \mathbb{R}$ with m vertices in general position, the problem of computing a canonical vertex permutation requires at least $\Omega(m \log m)$ time.*

Proof. We show the lower bound using a reduction from the problem of sorting a list of $M = \frac{m-2}{2}$ natural numbers. This problem is well-known to require at least $\Omega(m \log m)$ comparisons in worst case (cf. a classical book by Cormen *et al.* [58], Theorem 8.1). Let $a_1, \ldots, a_M$ be the elements of the list in order in which they appear in the list. We can determine the maximal element $a_{\max}$ in $O(M)$ time. We now construct a curve τ of complexity m as follows: let $x_i = 2i \cdot a_{\max}$, for $1 \leq i \leq M+1$, and let

$$\tau = 0,\ x_1,\ x_1 - a_1,\ \ldots,\ x_i,\ x_i - a_i,\ \ldots, x_M, x_M - a_M, x_{M+1}.$$

The constructed curve contains an edge between x_i and $x_i - a_i$ of length a_i, for every a_i of our sorting instance. We call these edges *variable edges*. The remaining edges of the curve τ are called *connector edges*. All connector edges are longer than $a_{\max}$. We can consider all the edges that would occur after extracting the two vertices of variable edges to be connector edges.

If the claim of the lemma would not hold, a canonical vertex permutation of τ would provide us the variable edges (i.e. the values x_i associated with these edges) in ascending order of their length, a contradiction. □

The following lemma testifies that we can query the canonical vertex permutation for a signature of a given size ℓ. Note that a canonical signature of size exactly ℓ may not exist.

Lemma 3.38. *Given a canonical vertex permutation of a curve τ, we can extract the canonical signature of τ of maximal size ℓ', with $\ell' \leq \ell$, in $O(\ell \log \ell)$ time.*

Proof. Let L' denote the suffix of the canonical vertex permutation which contains the last ℓ vertices. If there is no token separator at the starting position of L', then we remove the maximal prefix of L' which does not contain a token separator. In this way, we obtain the vertices of the canonical signature σ of maximal size ℓ', with $\ell' \leq \ell$. We now sort the vertices in L' in order of their appearance along σ in time $O(\ell \log \ell)$, and return the resulting curve. □

The following theorem follows from Lemma 3.36 and Lemma 3.38. Furthermore, Lemma 3.37 testifies that the preprocessing time is asymptotically tight.

Theorem 3.39. *Given a curve $\tau : [0,1] \to \mathbb{R}$ with m vertices in general position, we can construct a data structure in time $O(m \log m)$ and space $O(m)$, such that, given a parameter $\ell \in \mathbb{N}$ we can extract a canonical signature of maximal size ℓ', with $\ell' \leq \ell$, in time $O(\ell \log \ell)$.*

3.4.2 Computing signatures of a given error

Up to now, we have considered the problem of constructing a signature containing at most ℓ vertices, for a given value of $\ell \in \mathbb{N}$. Now we consider the construction of a δ-signature for a given parameter $\delta > 0$. Algorithm 1 does this, as it is claimed by Theorem 3.40. This algorithm is a greedy one, choosing the next signature vertex as the one that is out of the covered range, while being the farthest point in the allowed direction, such that the properties of Definition 3.3 are satisfied. The variable a counts the indices of the signature σ (as they are discovered), while the variable b counts the indices of the curve τ seen so far.

Algorithm 1: Computing a δ-signature

Data: curve $\tau = \tau(t_1), \ldots, \tau(t_m)$ with $0 = t_1 < \ldots < t_m = 1$, parameter $\delta > 0$
Result: values $s_1 < \ldots < s_\ell$, such that $\sigma = \tau(s_1), \tau(s_2), \ldots, \tau(s_\ell)$ is the δ-signature of τ

```
1  j ← 1; a ← 1; s_1 ← 0                          /* assign first vertex τ(0) to σ */
2  repeat j ← j + 1
3   until τ(t_j) ∉ [τ(0)]_δ or j ≥ m               /* scan beginning of the curve τ */
4  b ← j                                          /* τ(t_b) is first point outside [τ(0)]_δ */
5  for i ← j + 1 to m do                          /* scan remaining of the curve τ */
6      if τ(t_b) ∈ ⟨⟨τ(s_a), τ(t_i)⟩⟩ then        /* τ(t_i) farther than τ(t_b), and in the same
         direction */
7          b ← i                                  /* update farthest point of τ from τ(s_a) seen so far */
8      else
9          if |τ(t_i) − τ(t_b)| > 2δ then         /* gone backwards too far */
10             a ← a + 1; s_a ← t_b               /* append farthest point to σ */
11             b ← i                              /* update farthest point (and change direction) */
12 if τ(t_b) ∉ [τ(1)]_δ then                      /* check if the last vertex before τ(1) gets into σ */
13     a ← a + 1; s_a ← t_b;
14 a ← a + 1; s_a ← 1;                           /* assign last vertex τ(1) to σ */
15 return curve σ = τ(s_1), ..., τ(s_ℓ)
```

Theorem 3.40. *Given a curve $\tau : [0,1] \to \mathbb{R}$ with m vertices in general position, and given a parameter $\delta > 0$, Algorithm 1 computes a δ-signature $\sigma : [0,1] \to \mathbb{R}$ of τ in $O(m)$ time and using $O(m)$ space.*

Proof. We prove that Algorithm 1 produces the values $s_1 < \ldots < s_\ell$ that define a proper δ-signature $\sigma = \tau(s_1), \tau(s_2), \ldots, \tau(s_\ell)$ of τ according to Definition 3.3. The algorithm operates in three phases: (1) lines 2-4, (2) lines 5-11, and (3) lines 12-14. In the first phase, the algorithm finds the first vertex $\tau(t_j)$ of τ which lies outside the interval $[\tau(0)]_\delta$ (if such a vertex exists), and assigns its index to the variable b.

In the trivial case, τ is entirely contained in the interval $[\tau(0)]_\delta$. In this case, the first phase will simply run until the last vertex of τ ($j = m$), the second phase is not executed, and the condition in line 12 for the entry into the third phase evaluates to false. The algorithm returns the correct signature, which has two vertices, $s_1 = 0$ and $s_2 = 1$. Otherwise, τ must leave the interval $[\tau(0)]_\delta$, and there is the index $j < m$ of the first vertex $\tau(t_j)$ outside $[\tau(0)]_\delta$. For the rest of the proof, we assume that this happens.

We claim that the following invariants hold at the end of each iteration of the for-loop in the phase 2 (lines 5-11):

(I1) $\tau(s_1), \ldots, \tau(s_a)$ is a correct prefix of the δ-signature;
(I2) for any $x \in [s_a, t_i]$ it holds:
  (a) if $a > 1$ then $\tau(x) \in \langle\!\langle \tau(s_a), \tau(t_b) \rangle\!\rangle$,
  (b) if $a = 1$ then $\tau(x) \in [\tau(0) - \delta, \tau(t_b)]$ when $\tau(t_b) > \tau(0)$
    (respectively, $\tau(x) \in [\tau(t_b), \tau(0) + \delta]$ when $\tau(t_b) < \tau(0)$);
(I3) (a) if $a > 1$, then $|\tau(s_a) - \tau(t_b)| > 2\delta$,
  (b) if $a = 1$ then $|\tau(0) - \tau(t_b)| > \delta$;
(I4) if $t_i > t_b$, then for any $x \in [t_b, t_i]$ it is $|\tau(t_b) - \tau(x)| \leq 2\delta$;
(I5) the direction-preserving property holds for the subcurve $\tau[s_a, t_b]$.

We prove the invariants by induction on i (i.e. over the vertices of the curve τ). The base case happens after execution of line 4, before the first iteration of the for-loop (i.e. at the end of the zeroth iteration). For ease of notation, we define $i = b$ for this case. Invariants (I1), (I3) and (I4) hold by construction. The invariants (I2) and (I5) follow immediately from the observation that $\tau(t_b)$ is the first point outside the interval $[\tau(0)]_\delta$.

Now we prove the induction step. During the execution of the for-loop in lines 5-11, we implicitly maintain a general direction in which the curve τ is moving. This direction is *upwards* if $\tau(s_a) < \tau(t_b)$ and *downwards* otherwise. Furthermore, we maintain that $\tau(t_b)$ is the farthest point from $\tau(s_a)$ on the current signature edge (starting at $\tau(s_a)$) in the current general direction. Note that a new vertex is appended to the signature prefix (in

line 10) only when τ has already moved in the opposite direction by a distance greater than 2δ. Only then, we say that the current general direction of the curve has changed.

Consider an arbitrary iteration i of the for-loop. There are three cases:

(i) line 7 is executed and i becomes the new b
(this happens if τ is moving in the current general direction beyond $\tau(t_b)$);

(ii) lines 10 and 11 are executed and a new signature vertex is appended to the signature prefix
(this happens if τ has changed its general direction, as the new vertex satisfying minimum-edge-length property was found);

(iii) no assignments were made
(this happens if τ locally changes direction, but the current general direction does not change, i.e. $\tau(t_b) \notin \langle\langle \tau(s_a), \tau(t_i) \rangle\rangle$ and $|\tau(t_i) - \tau(t_b)| \leq 2\delta$).

For each invariant we consider each of the three cases above.

(I1) If the signature prefix was not changed in the previous iteration (cases (i) and (iii)), then (I1) simply holds by induction. Otherwise, we argue that the new signature prefix is correct. By induction, $\tau(s_1), \ldots, \tau(s_{a-1})$ is a correct signature prefix. The conditions of Definition 3.3 for $\tau(s_1), \ldots, \tau(s_a)$ follow by the induction hypothese in the iteration step $i' < i$, in which the last value of b was assigned. In particular, (i) non-degeneracy follows from (I2), (ii) direction-preserving follows from (I5), (iii) minimum-edge-length follows from (I3), and (iv) range property follows from (I2).

(I2) We distinct the two cases:

$a = 1$**:** Let $\tau(t_b) > \tau(0)$. Since $a = 1$, we cannot be in case (ii). Furthermore, once we enter the for-loop, the current general direction is fixed until a is incremented for the first time. Therefore, by (I2) in the $(i-1)$-th iteration, we have for $x \in [s_1, t_{i-1}]$ that $\tau(x) \in [\tau(0) - \delta, \tau(t_{b'})]$, where b' holds the value of b before we entered the for-loop in the current iteration. Now, in case (i) the claim follows immediately. In case (iii) it follows from the (false) condition in line 9, that $\tau(t_i) > \tau(t_b) - 2\delta \geq \tau(0) - \delta$, and by the (false) condition in line 6, that $\tau(t_i) < \tau(t_b)$. The case $\tau(t_b) < \tau(0)$ is analogous.

$a > 1$**:** Assume case (ii) and assume that the general direction changed from upwards to downwards (the opposite case is analogous). Let a' and b' be the values of a and b before the new assignment in lines 10 and 11. By (I2) in the iteration when one of these values were previously assigned, we have $\tau(t_{b'}) \geq \tau(x)$ for any $x \in [t_{b'}, t_{i-1}]$. By (I4), we have $\tau(t_{b'}) - 2\delta \leq \tau(x)$ for any $x \in [t_{b'}, t_{i-1}]$. By the (true) condition in line 9, we have $\tau(t_i) < \tau(t_{b'}) - 2\delta$. Therefore, for any

$x \in [t_{b'}, t_i]$, we have $\tau(x) \in [\tau(t_i), \tau(t_{b'})]$, which implies (I2) after the assignment in lines 10 and 11.

Now, in case (i) and case (iii), we have by (I2) (in the previous iteration) for $x \in [s_a, t_{i-1}]$ that $\tau(x) \in \langle\!\langle \tau(s_a), \tau(t'_b) \rangle\!\rangle$. The correctness in case (i) follows immediately. In case (iii), assume $\tau(t_b) > \tau(s_a)$. It follows from the (false) condition in line 9 and by (I3), that $\tau(t_i) > \tau(t_b) - 2\delta \geq \tau(s_a)$, and by the (false) condition in line 6, that $\tau(t_i) < \tau(t_b)$. The case $\tau(t_b) < \tau(s_a)$ is analogous.

(I3) Again, we distinguish the two cases:

$i = 1$**:** since the for-loop was started after the curve τ left the interval $[\tau(0)]_\delta$ for the first time, and by (I2) in the previous iteration, $\tau(t_b)$ is farthest point from $\tau(0)$, and thus, the invariant (I3) remains valid.

$a > 1$**:** In case (ii), we append the parameter t_b to the signature prefix and re-initialize b to be i only after the curve has moved by a distance of at least 2δ (line 9) from $\tau(t_b)$, thus (I3) is valid. For the cases (i) and (iii), the distance (inherited by (I3) from the previous iteration) is further maintained by (I2).

(I4) is clearly satisfied in case (i) and case (ii), since $b = i$ is assigned. In case (iii), when no new assignment is done, for the curve $\tau[t_b, t_{i-1}]$, (I4) follows by induction from the previous iteration. For the curve $\tau[t_{i-1}, t_i]$, (I4) holds by the (false) condition in line 9 that $|\tau(t_b) - \tau(t_i)| \leq 2\delta$.

(I5) holds since we assign a new signature vertex with parameter $s_a = t_b$ as soon as the curve moves by more than 2δ in the opposite direction (case (ii)). In the other two cases there is no change, thus the correctness follows by induction from the previous iteration.

By induction, we conclude from the invariant (I1) that after the phase 2 of the algorithm, the vertices chosen into σ make a correct prefix of the δ-signature. It remains to decide on the last two vertices. In phase 3, there are two cases. If the range condition is satisfied for the edge from $\tau(s_a)$ to $\tau(1)$ (potentially the last signature edge), the algorithm only appends the last vertex $\tau(1)$ of the curve τ to the signature σ. Otherwise, the algorithm appends $\tau(t_b)$ and $\tau(1)$ to the signature, since we have moved from $\tau(t_b)$ in the opposite direction by more than δ, and by (I3), we had previously a signature edge of length more than 2δ. In both cases, the conditions in Definition 3.3 are satisfied for the part of the curve we considered in phase 3 as well.

We can use a linked list to store the parameters of the vertices of the signature. Then, the running time and the space requirements of the algorithm are linear in m, since the

execution of one iteration of the for-loop takes constant time, and there are at most m such iterations. This closes the proof of the theorem. □

3.4.3 Signatures as approximate minimum-error simplification

We close this section with the result that shows that signatures are a 2-approximation to the minimum-error ℓ-simplification problem (cf. Definition 3.1).

Lemma 3.41. *Given a curve $\tau : [0,1] \to \mathbb{R}$ with m vertices in general position, and given a parameter $\ell \in \mathbb{N}$, we can compute in $O(m \log m)$ time a curve $\pi : [0,1] \to \mathbb{R}$ with at most ℓ vertices, such that $d_F(\pi, \tau) \leq 2d_F(\pi^*, \tau)$, for π^* being a minimum-error ℓ-simplification of τ.*

Proof. Let $\sigma_1, \ldots, \sigma_k$ be the canonical signatures of τ with corresponding parameters $\delta_1, \ldots, \delta_k$, as defined in Lemma 3.34. Lemma 3.4 implies that $d_F(\sigma_i, \tau) \leq \delta_i$. Consider the signature σ_i with the maximal number of $\ell' \leq \ell$ vertices. We claim that

$$\frac{\delta_i}{2} \leq d_F(\pi^*, \tau) \leq \delta_i,$$

which will imply the claim of the lemma. The second inequality follows from $d_F(\pi^*, \tau) \leq d_F(\sigma_i, \tau) \leq \delta_i$, by the definition of π^* and the fact that σ_i consists of at most ℓ vertices. To see the first inequality, consider the signature $\sigma_{i-1} = v_1, \ldots, v_h$, with $h > \ell$. By Lemma 3.34(i), the signature σ_{i-1} is a δ-signature of τ for all $\delta \in [\delta_{i-1}, \delta_i)$, and so for $\delta = \delta_i - \varepsilon$, for any $\varepsilon > 0$. By Definition 3.3, it holds for

$$R_j = \left[v_j - \frac{\delta}{2}, v_j + \frac{\delta}{2} \right],$$

that for any $1 \leq j \leq h-1$ it is $R_j \cap R_{j+1} = \emptyset$.

Repeating the proof of Lemma 3.5, but with ranges $R_j = [v_j - \delta/2, v_j + \delta/2]$ instead of $[v_j - \delta, v_j + \delta]$ (the rest of the proof can be taken verbatim), we conclude that any curve π with $d_F(\pi, \tau) \leq \delta/2$ needs to consist of at least $h > \ell$ vertices. Since π^* has complexity at most ℓ, the first inequality follows. We can compute the signature σ_i in $O(m \log m)$ time using Theorem 3.39. This closes the proof of the lemma □

3.5 Conclusion and open questions

In this chapter we introduced the special type of the curve simplification in one-dimensional ambient space – the δ-signatures. The advantage of signatures is that for any input curve of complexity m, they can be efficiently computed, both if an error threshold δ, or a

goal size ℓ is given (in time $O(m)$ and $O(m \log m)$, respectively). The signatures yield a 2-approximative solution to the minimum-error ℓ-simplification problem on the input curve τ. Thus, the signatures can be incorporated as a tool into algorithms that require an ℓ-simplification, without requiring a large computation time.

Once we computed a δ-signature of a given input curve τ, we can search for vertices of the curves that are similar to τ, i.e. at continuous Fréchet distance to τ at most δ, only in the intervals of width 2δ around the signature vertices. All other possible vertices can be "forgotten", since we showed that by removing such a vertex from a curve that is close to τ does not increase the distance of the curve beyond δ.

A strict restriction to the application of the signatures is the dimension of their ambient space. It would be great to have an analogous concept in higher dimensional spaces, or at least in the two-dimensional space. However, it is not clear how to formulate such a simplification concept, in order to be able to repeat the proof of the crucial Lemma 3.7. This remains the main open question of this chapter.

4 Clustering under the Fréchet distance

4.1 Introduction

Clustering of curves is an active research topic, both in algorithmic theory and in data mining community, with results dating back to the well-known Lloyd's algorithm from 1957 [132], which was developed for the k-means clustering. However, most of the approaches in data mining, where the notion of *time series* instead of *curves* is rather used, lack a rigorous algorithmic analysis. For an overview of data mining approaches and methods we confer to the surveys of Aghabozorgi *et al.* [10], Jacques and Preda [114], and Liao [131]. For a more extensive discussion we refer to the book chapter by Kotsakos *et al.* [121] and references therein.

A common approach to the curve analysis is to observe each vertex of the input curves as a coordinate (or a tuple of coordinates) and thus a curve as a high-dimensional point (cf. [131]). Upon such interpretation of data, any clustering algorithm can be applied. Despite the practicality of such an interpretation and its simplicity, there are at least two main drawbacks that are often hard to resolve: that all curves must be of the same complexity, and if the vertices of the curves have embedded the time aspect, then these need to be synchronized.

When choosing a single representative for a set of curves under Fréchet distance, the optimal solution may have a complexity that equals the sum of the complexities of the input curves (cf. [12]). This can cause a vast overfitting. Therefore, we opt to adapt the classic k-clustering problems (cf. Section 2.4) by bounding the complexity of the clustering center curves by a constant $\ell \in \mathbb{N}$.

4.1.1 Problem definition

We define our problems for curves in the one-dimensional Euclidean ambient space. Remember that Δ_m denotes the set of the polygonal curves of complexity m in the ambient space $\mathbb{R}$ (cf. to the page 39 for the definitions of Δ and Δ_m). The formulations of Equa-

tions (4.1) and (4.2) can easily be extended for the curves in $\mathbb{R}^d$. As in Chapter 3, we make the general position assumption[11] on the input curves.

Given a set of n curves $P = \{\tau_1, \ldots, \tau_n\} \subseteq \Delta_m$ and parameters $k, \ell \in \mathbb{N}$ that we assume to be constants, we define a (k, ℓ)-**clustering** as a set of k curves $C = \{\varsigma_1, \ldots, \varsigma_k\}$ taken from Δ_ℓ which minimize one of the following cost functions:

$$\mathrm{cost}_\infty(P, C) = \max_{i \in \{1,\ldots,n\}} \min_{j \in \{1,\ldots,k\}} d_F(\tau_i, \varsigma_j)\,, \tag{4.1}$$

$$\mathrm{cost}_1(P, C) = \sum_{i=1}^{n} \min_{j \in \{1,\ldots,k\}} d_F(\tau_i, \varsigma_j)\,. \tag{4.2}$$

In case that we have only one clustering center curve ς (when $k = 1$), we will simply write $\mathrm{cost}_i(P, \varsigma)$ instead of $\mathrm{cost}_i(P, \{\varsigma\})$, for $i \in \{\infty, 1\}$. We refer to the clustering problem as (k, ℓ)-**center** (Equation 4.1) and (k, ℓ)-**median** (Equation 4.2), respectively. We define the cost of an optimal solution as

$$\mathrm{opt}^{(i)}_{k,\ell}(P) = \min_{C \subset \Delta_\ell} \mathrm{cost}_i(P, C),$$

where the restrictions on C are as described above and $i \in \{\infty, 1\}$.

We define and analyze both problems using continuous Fréchet distance. It is of independent interest to study both problems under discrete Fréchet distance as well. Since parts of our analysis for the continuous case work for the discrete Fréchet distance too, we discuss the differences and respective results at the end of each section.

4.1.2 Results in this chapter

Let $0 < \varepsilon < 1$. We present the first $(1 + \varepsilon)$-approximation algorithms for the (k, ℓ)-center (cf. Theorem 4.13) and (k, ℓ)-median (cf. Theorem 4.22) problem under the continuous Fréchet distance, for the curves in the one-dimensional Euclidean space. Both algorithms produce a witness solution and a $(1 + \varepsilon)$-approximate cost for the respective problems, in time $\tilde{O}(mn)$.

In order to produce a solution for (k, ℓ)-median, we have to overcome the problem that the metric spaces we work in do not have the bounded doubling dimension, neither for the continuous nor for the discrete Fréchet distance. We show these facts first in Section 4.2.

Both of our $(1 + \varepsilon)$-approximation algorithms use the properties of signatures, presented in Chapter 3. Unfortunately, the signatures cannot be used for the discrete Fréchet distance

[11] See page 49 for the definition.

case. Therefore, for the (k, ℓ)-center and the (k, ℓ)-median problems under the discrete Fréchet distance we present constant-factor approximation algorithms, with approximation factors of 5 and 45, respectively. The running times of these algorithms are $O(mn)$. We state these results as Theorem 4.10 and Theorem 4.11. Note that both of these results are valid for any dimension $d \geq 1$ of the ambient space.

We also show that both problems are **NP**-hard, if k is part of the input, for $\ell \geq 2$, under both continuous and discrete Fréchet distance. The (k, ℓ)-center problem is **NP**-hard to approximate better than a factor of 2. Theorem 4.25 and Theorem 4.26 show these results.

4.1.3 Related work

Until recently not so much was known on computing of curves' clustering under dissimilarity measures that do not treat curves as sets. Dumitrescu and Rote [72] considered the problem of simultaneous minimization of the Fréchet distance between all pairs of curves from a set[12] of n curves of complexity m, and provided a 2-approximation solution in time $O(m^n \log m)$. For this problem (under the discrete Fréchet distance), Buchin *et al.* [39] showed that unless SETH (cf. Hypothesis 2.34) fails, the solution cannot be computed significantly faster than $O(m^n)$.

Searching for a single representative of a set of n input curves was the first clustering-like approach. Buchin *et al.* [41] defined the median level curve using only parts of the input curves. Har-Peled and Raichel [98] defined a mean curve, which minimizes the distance to the input curves, and which can be chosen with no restrictions. Ahn *et al.* [12] defined the middle curve problem, which uses only vertices of the input curves and minimizes the Fréchet distance to the input. Both of these algorithms need exponential time in the number of the input curves n.

On clustering of curves with multiple representative curves there were no published results before our work, which was published in [68], and which is presented in this chapter. These results started a series of publications that we consider in the following.

For the (k, ℓ)-center problem it was later shown by Buchin *et al.* [44] that if ℓ is part of the input, then there is no polynomial time approximation scheme. They reduced the problem to the Shortest Common Supersequence (SCS) problem. The approximation factor bound depends on the dimension of the ambient space d and on whether the Fréchet distance is discrete or continuous. These lower bound factors are presented in Table 4.1, and are originally from [44]. These bounds hold even if $k = 1$. The (k, ℓ)-median problem is **NP**-hard as well, if ℓ is part of the input. This was shown by Buchin, Driemel and Struijs [45] by reduction to the SCS problem.

[12] They called this the Fréchet distance of the set of curves.

	Continuous Fréchet distance	Discrete Fréchet distance
$d = 1$	1.5	2
$d \geq 2$	2.25	2.598

Table 4.1: The lower bounds for the approximation factor of an approximation algorithm for the (k, ℓ)-center problem, if ℓ is part of the input. These results were presented in [44].

On the positive side, there exists a constant-factor approximation algorithm for $d \geq 2$ by Buchin *et al.* [44]. They adapted the algorithm of Gonzalez [89], with approximation factor of 3 for the discrete Fréchet distance (in time $\tilde{O}(mn)$), and the factors 3 and 6 for $d = 2$ and $d > 2$ respectively, for the continuous Fréchet distance (in time $\tilde{O}(mn + m^3)$). The result for the discrete Fréchet distance was later improved by Buchin, Driemel and Struijs [45] into a $(1 + \varepsilon)$-approximation algorithm with running time $\tilde{O}(mn)$. They also gave an exact algorithm for $d \leq 2$ with running time $\tilde{O}((mn)^{2k\ell+1})$.

For the (k, ℓ)-median problem an improvement to our result was given by Buchin, Driemel, and Struijs [45]. They gave a $(1 + \varepsilon)$-approximation algorithm for $d > 1$ under the discrete Fréchet distance in time $\tilde{O}(nm^{dkl+1})$. This result was further improved into a $(1 + \varepsilon)$-approximation algorithm under discrete Fréchet distance by Nath and Taylor [146], with running time $\tilde{O}(mn)$. Their approach extends to the k-median under Hausdorff distance.

To find a $(1 + \varepsilon)$-approximation to the (k, ℓ)-median clustering under the continuous Fréchet distance for $d > 1$ is still an open problem. However, for $d > 1$ there are recent results by Meintrup, Munteanu and Rohde [139], and by Buchin, Driemel and Rohde [45], that both obtain a $(1 + \varepsilon)$-approximation solution to the (k, ℓ)-median clustering under the continuous Fréchet distance, but with a caveat. The result of Meintrup, Munteanu and Rohde [139] assumes that the number of outlier input curves is bounded. The result of Buchin, Driemel and Rohde [46] has no assumptions on the input, but yields a bicriteria approximation solution with complexity of each center curve at most $2\ell - 2$, in time linear in n and polynomial in m.

We summarize the best known results for the problems we consider in this thesis in Table 4.2.

For the (k, ℓ)-means problem (an extension of the known k-means problem analogous to our extensions of the k-center and the k-median problems) and for the (k, ℓ)-clustering problem under the DTW distance there are no known results in the literature.

On problems related to the (k, ℓ)-clustering the following is known. When the restrictions on ℓ are lapsed, we have classical k-clustering problems: 1-center (smallest enclosing ball)

		Continuous Fréchet distance			Discrete Fréchet distance		
(k,ℓ)-**center**	$d=1$	Theorem 4.13	$1+\varepsilon$	$\tilde{O}(mn)$	[45]	$1+\varepsilon$	$\tilde{O}(mn)$
	$d=2$	[44]	3	$\tilde{O}(mn+m^3)$	[45]	$1+\varepsilon$	$\tilde{O}(mn)$
	$d>2$	[44]	6	$\tilde{O}(mn+m^3)$	[45]	$1+\varepsilon$	$\tilde{O}(mn)$
(k,ℓ)-**median**	$d=1$	Theorem 4.22	$1+\varepsilon$	$\tilde{O}(mn)$	[146]	$1+\varepsilon$	$\tilde{O}(mn)$
	$d\geq 2$	?	?	?	[146]	$1+\varepsilon$	$\tilde{O}(mn)$
bicriteria	$d\geq 2$	[46]	$1+\varepsilon$	$O\left(m^{O(1)}n\right)$			

Table 4.2: The best known approximation algorithms for the (k,ℓ)-center and the (k,ℓ)-median problems. For each result the reference, the approximation factor, and the running time are given.

and 1-median problems are **NP**-hard under both discrete and continuous Fréchet distance (cf. [45]). 1-median under DTW (and its variants) are **NP**-hard as well, where the result of [45] generalizes the previous result by Bulteau, Froese, and Niedermeier [49]. Based on technique of [44, 45], Buchin, Funk, and Krivošija [48] showed that the middle curve problem of Ahn *et al.* [12] is **NP**-hard.

4.2 Doubling dimension of the metric space

The standard clustering techniques [3, 126] for metric spaces with bounded doubling dimension cannot be directly applied to (k,ℓ)-clustering problems under (continuous and discrete) Fréchet distance. Namely, the doubling dimension of the space of one-dimensional curves in $\mathbb{R}$, i.e. univariate time series, is unbounded. This result is presented by Lemma 4.1. Even if we restrict the complexity of the curves to $\ell \geq 4$, the doubling dimension is unbounded, which is claimed by Lemma 4.2. Note that Lemma 4.2 does not hold under the discrete Fréchet distance. We discuss this at the end of the section, together with cases when $\ell < 4$.

Lemma 4.1. *The doubling dimension of the metric spaces (Δ, d_F) and (Δ, d_{dF}) is unbounded.*

Proof of Lemma 4.1 for the continuous Fréchet distance. Assume for the sake of contradiction that the doubling dimension of (Δ, d_F) is bounded and equal to some $d \in \mathbb{N}$. We construct an instance of $2^d + 1$ curves from Δ, which lie in a ball of radius $\frac{1}{8}$ while no

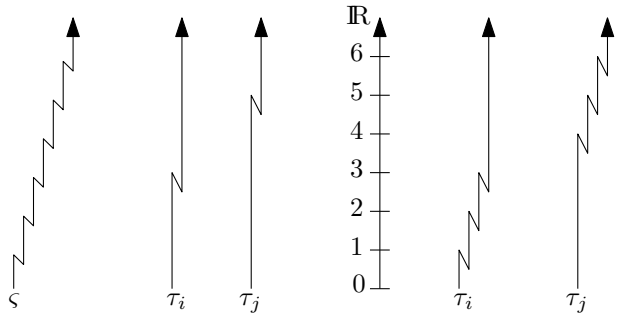


Figure 4.1: Examples of the constructed curves in Lemma 4.1 (left) and Lemma 4.2 (right).

two elements can be covered by a ball of radius $\frac{1}{16}$. But, by pigeonhole principle there must exist two curves from the input set, such that they are covered by one ball of radius $\frac{1}{16}$.

Let therefore be $P = \{\tau_1, \ldots, \tau_{2^d+1}\}$, with $\tau_i = 0, i, i - \frac{1}{2}, 2^d + 2$, for $i \in [2^d + 1]$. The set P is contained in the ball $\mathbf{B}\big(\varsigma, \frac{1}{8}\big)$, where ς is the curve

$$\varsigma = 0, \frac{7}{8}, \frac{5}{8}, \ldots, i - \frac{1}{8}, i - \frac{3}{8}, \ldots, 2^d + \frac{7}{8}, 2^d + \frac{5}{8}, 2^d + 2.$$

An example is given in Figure 4.1 (left).

Any two curves $\tau_i, \tau_j \in P$ have Fréchet distance $\frac{1}{4}$ to each other. Now, assume that a ball of radius $\frac{1}{16}$ exists which contains both τ_i and τ_j. Let its center be denoted ς_{ij}. Using the triangle inequality we get:

$$\frac{1}{4} = d_F(\tau_i, \tau_j) \leq d_F(\tau_i, \varsigma_{ij}) + d_F(\varsigma_{ij}, \tau_j) \leq \frac{1}{8},$$

a contradiction. □

Proof of Lemma 4.1 for the discrete Fréchet distance. We follow the proof for the continuous case. We adapt the curves of the input set P and the center curve by adding vertices along the curves without changing the shape of the curves. Let $\tau_i = 0, 1 - \frac{1}{4}, \ldots, i - 1 - \frac{1}{4}, i, i - \frac{1}{2}, i + \frac{3}{4}, \ldots, 2^d + \frac{3}{4}, 2^d + 2$, and let the center curve be

$$\varsigma = 0, \frac{7}{8}, \frac{6}{8}, \frac{5}{8}, \ldots, i - \frac{1}{8}, i - \frac{2}{8}, i - \frac{3}{8}, \ldots, 2^d + \frac{7}{8}, 2^d + \frac{6}{8}, 2^d + \frac{5}{8}, 2^d + 2.$$

The rest of the proof holds verbatim, since the discrete Fréchet distance between each two of the input curves is $\frac{1}{4}$. □

Lemma 4.2. *For any integer $\ell \geq 4$, the doubling dimension of the metric space (Δ_ℓ, d_F) is unbounded.*

Proof. The proof is similar to the proof of Lemma 4.1. This time we argue that no two curves in the set can be covered by a ball of half the radius because there exists no suitable center in Δ_ℓ, that is, the center would need to have complexity higher than ℓ. As in the proof of Lemma 4.1, we define a set $P = \{\tau_1, \ldots, \tau_{2^d+1}\} \subset \Delta_\ell$. For $s = \lfloor \frac{\ell-2}{2} \rfloor$ and $i \in [2^d + 1]$, let

$$\begin{aligned}\tau_i = \quad & 0, s(i-1)+1, s(i-1)+\frac{1}{2}, \ldots, s(i-1)+j, s(i-1)+j-\frac{1}{2}, \ldots \\ & \ldots, s(i-1)+s, s(i-1)+s-\frac{1}{2}, s(2^d+2),\end{aligned}$$

where $j \in [s]$. Note that if ℓ is odd, the curve τ_i has $\ell - 1$ vertices. Clearly, each $\tau_i \in P$ is an element of Δ_ℓ, since its complexity is at most ℓ. Figure 4.1 (right) presents an example with $\ell = 8$ (and $s = 3$). Note that the choice of the curves τ_i requires $\ell \geq 4$, otherwise our curves would have only one vertex 0.

The set P is contained in the ball with radius $\frac{1}{4}$ centered at $\varsigma = 0, s(2^d + 2)$. Note that the $\frac{1}{8}$-signature of any $\tau_i \in P$ is equal to τ_i itself. Thus, by Lemma 3.5, any potential center curve ς_i of a ball of radius $\frac{1}{8}$, such that $\tau_i \in \mathbf{B}\big(\varsigma_i, \frac{1}{8}\big)$ needs to have a vertex in each interval $\big[w - \frac{1}{8}, w + \frac{1}{8}\big]$ for each vertex w of τ_i. By construction, these intervals are pairwise disjoint for each curve and across all curves in P (except for the intervals around the two endpoints). Therefore, such a ball with radius $\frac{1}{8}$ which would cover two different curves $\tau_i, \tau_j \in P$, would need to have the center curve with more than ℓ vertices and is therefore not contained in Δ_ℓ. Indeed, the number of pairwise disjoint signature intervals induced by any $\tau_i, \tau_j \in P$ with $i \neq j$, is $2 + 2 \cdot 2s \geq 2\ell - 2 > \ell$, since $\ell \geq 4$. □

We would like to have an analogous claim as Lemma 4.2 under discrete Fréchet distance, but in that case the doubling dimension of the metric space (Δ_ℓ, d_{dF}) is bounded. We show this in the following lemma.

Lemma 4.3. *For any integer $\ell \geq 2$, the doubling dimension of the metric space (Δ_ℓ, d_{dF}) is bounded.*

Proof. Let $\varsigma = c_1, \ldots, c_\ell$ be a curve in Δ_ℓ, and let $r > 0$. Each curve $\tau \in \mathbf{B}(\varsigma, r)$, must have a vertex in each of the ranges $[c_i - r, c_i + r]$, for all $i \in [\ell]$, otherwise a traversal that realizes $d_{dF}(\varsigma, \tau) \leq r$ would not exist. The curve τ we assign to a curve $\varsigma_j = c_{j1}, \ldots, c_{j\ell}$ in the following manner. Let w_i be a vertex of τ that lies in the range $[c_i - r, c_i + r]$, for $i \in [\ell]$. Then, the vertex $c_{ji} = c_i - \frac{r}{2}$ if $w_i \in [c_i - r, c_i)$, and $c_{ji} = c_i + \frac{r}{2}$ otherwise (i.e. if $w_i \in [c_i, c_i + r]$). There are 2^ℓ possible assignments, and thus at most 2^ℓ distinct curves ς_j. Clearly $\tau \in \mathbf{B}\big(\varsigma_j, \frac{r}{2}\big)$, therefore the ball $\mathbf{B}(\varsigma, r)$ is covered with at most 2^ℓ balls of half the radius. This closes the proof. □

Under continuous Fréchet distance for small values of ℓ (i.e. $\ell \in \{2, 3\}$) we can analogously to Lemma 4.3 conclude that the doubling dimension of the space (Δ_ℓ, d_F) is bounded, since for each center curve from Δ_2 (respectively Δ_3) and any radius $r > 0$, we can cover the ball $\mathbf{B}(\varsigma, r)$ with at most 2^ℓ balls of half the radius. Note that for $\ell = 2$ the metric space (Δ_ℓ, d_F) equals the metric space $(\mathbb{R}^2, \ell_\infty)$, which is known to have a bounded doubling dimension (cf. Gupta, Krauthgamer, Lee [93]).

4.3 Constant-factor approximation

A constant-factor approximation algorithm for a clustering problem is often a first step in construction of a $(1 + \varepsilon)$-approximation. The algorithm we present, first simplifies the input curves while reducing their complexity to a constant, and then applies an existing k-clustering approximation algorithm designed for general metric spaces. One aspect we need to take care of for the choice of algorithms is that our clustering centers have to have complexity ℓ. This can be obtained if the used algorithms return a subset of the input, whose curves are in Δ_ℓ.

For k-center problem we use the algorithm of Gonzalez [89], here stated as in [95]. This algorithm is a greedy algorithm, it selects an arbitrary input point as the first center, and subsequently $k - 1$ times selects and adds into set of centers the point from the input that is at maximum distance to the so far selected centers. This result we state as Theorem 4.4.

Theorem 4.4 ([95] Theorem 4.3)**.** *Given a set P of n points in a metric space $(\mathcal{X}, \mathbf{d})$, there is an algorithm that computes a set of k centers, which is a subset of P, such that it is a 2-approximation to the optimal k-center clustering of P. The running time is $O(nk \cdot T(\mathcal{X}))$ time, where $T(\mathcal{X})$ is the time needed to compute distance $\mathbf{d}$ between a pair of points in $\mathcal{X}$.*

Chen [54] has given a constant-factor approximation algorithm for the discrete k-median problem in general metric spaces, i.e. when the cluster centers are selected among the input points. Chen's algorithm begins with the construction of a strong coreset Q (cf. Definition 2.28) of the input set P for k-median problem, whose size depends only logarithmically on n. This construction is probabilistic, chooses the points among the elements of P, and guarantees success with constant probability. Running the local search algorithm of Arya *et al.* [16] on Q yields a $(5 + \varepsilon)$-approximative k-median solution for Q. Since the local search algorithm explores the elements of Q as potential solution centers, the found solution is a subset of Q, therefore it is a subset of P as well. Chen proved that it is a $(10 + \varepsilon)$-approximation to the optimal solution of the k-median problem on P. This result we state as Theorem 4.5.

Theorem 4.5 ([54] Theorem 6.2)**.** *Given a set P of n points in a metric space, for $0 < \varepsilon < 1$, there is an algorithm that computes a set of k centers, which is a subset of P, such that it is a $(10 + \varepsilon)$-approximation to the optimal k-median clustering of P, with constant probability of success. The running time is $O\left(nk + k^7\varepsilon^{-5}\log^5 n\right)$.*

The following algorithm provides a constant-factor approximation to the (k, ℓ)-clustering problems.

Algorithm 2: Constant-factor approximation for (k, ℓ)-clustering

Data: curves $P = \{\tau_1, \ldots, \tau_n\}$, parameters $k > 0, \ell > 0$
Result: cluster centers $C = \{\varsigma_1, \ldots, \varsigma_k\}$ and cost D
1 For each τ_i compute an approximate minimum-error ℓ-simplification $\widehat{\tau}_i$ (Lemma 3.41)
2 Apply an approximation algorithm for k-clustering in general metric spaces on $\widehat{P} = \{\widehat{\tau}_1, \ldots, \widehat{\tau}_n\}$ (Gonzales' algorithm (Theorem 4.4) for k-center and Chen's algorithm (Theorem 4.5) for k-median)
3 **return** the resulting cluster centers $C = \{\varsigma_1, \ldots, \varsigma_k\}$ with approximate cost

$$\begin{aligned} D_\infty &= \mathrm{cost}_\infty(C, \widehat{P}) + \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \\ D_1 &= \mathrm{cost}_1(C, \widehat{P}) + \sum_{i=1}^{n} d_F(\tau_i, \widehat{\tau}_i) \end{aligned}$$

for (k, ℓ)-center and (k, ℓ)-median, respectively

Lemma 4.6. *The cost D_∞ (respectively, D_1) and the solution C computed by Algorithm 2 constitute a $(\alpha + \beta + \alpha\beta)$-approximation to the (k, ℓ)-center problem (respectively, the (k, ℓ)-median problem), where α is the approximation factor of the simplification step and β is the approximation factor of the clustering step.*

Proof. We first discuss the case of (k, ℓ)-center. We have that

$$\begin{aligned} D_\infty &= \mathrm{cost}_\infty(C, \widehat{P}) + \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \\ &= \max_{i\in\{1,\ldots,n\}} \min_{\varsigma\in C} d_F(\widehat{\tau}_i, \varsigma) + \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \\ &\geq \max_{i\in\{1,\ldots,n\}} \left(\min_{\varsigma\in C} d_F(\widehat{\tau}_i, \varsigma) + d_F(\tau_i, \widehat{\tau}_i)\right) \\ &\geq \max_{i\in\{1,\ldots,n\}} \min_{\varsigma\in C} (d_F(\widehat{\tau}_i, \varsigma) + d_F(\tau_i, \widehat{\tau}_i)) \\ &\geq \max_{i\in\{1,\ldots,n\}} \min_{\varsigma\in C} d_F(\tau_i, \varsigma) \\ &\geq \mathrm{cost}_\infty(C, P). \end{aligned}$$

Let δ^* be the optimal cost for a solution to the (k,ℓ)-center problem for $P = \{\tau_1, \ldots, \tau_n\}$, and let one such optimal solution be C^*. If we denote the optimal ℓ-simplification of τ_i with $\widehat{\tau}_i^*$, then we have

$$\max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \leq \alpha \cdot \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i^*) \leq \alpha \cdot \max_{i\in\{1,\ldots,n\}} \min_{\omega\in C^*} d_F(\tau_i, \omega) = \alpha\delta^*,$$

since $d_F(\tau_i, \omega)$, $\omega \in C^*$, is lower bounded by the Fréchet distance of the input time series τ_i to its optimal ℓ-simplification, as the curves $\omega \in C^*$ have at most ℓ vertices. Thus it holds that

$$D_\infty \quad \leq \quad \mathrm{cost}_\infty(C, \widehat{P}) + \alpha\delta^*.$$

To relate an optimal solution C^* to D_∞, we proceed as follows. Let $\omega_i \in C^*$ be the center of this optimal solution which is closest to τ_i, for $1 \leq i \leq n$. We have

$$\begin{aligned}
\delta^* &= \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \omega_i) \\
&\geq \max_{i\in\{1,\ldots,n\}} \left(d_F(\widehat{\tau}_i, \omega_i) - d_F(\tau_i, \widehat{\tau}_i)\right) \\
&\geq \max_{i\in\{1,\ldots,n\}} d_F(\widehat{\tau}_i, \omega_i) - \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \\
&\geq \mathrm{cost}_\infty(C^*, \widehat{P}) - \max_{i\in\{1,\ldots,n\}} d_F(\tau_i, \widehat{\tau}_i) \\
&\geq \frac{1}{\beta}\Big(\mathrm{cost}_\infty(C, \widehat{P})\Big) - \alpha\delta^* \\
&\geq \frac{1}{\beta}(D_\infty - \alpha\delta^*) - \alpha\delta^*,
\end{aligned}$$

since

$$\mathrm{cost}_\infty(C, \widehat{P}) \leq \beta \cdot \mathrm{cost}_\infty(\widehat{C}^*, \widehat{P}) \leq \beta \cdot \mathrm{cost}_\infty(C^*, \widehat{P}),$$

where $\widehat{C}^*$ is an optimal solution to the (k,ℓ)-center problem for $\widehat{P}$. We conclude that $\delta^* \leq \mathrm{cost}_\infty(C, P) \leq D_\infty \leq (\alpha + \beta + \alpha\beta)\delta^*$, as claimed.

The claim of the lemma for the (k,ℓ)-median problem follows with small modifications, that we give here for completeness. We have that

$$\begin{aligned}
D_1 &= \mathrm{cost}_1(C, \widehat{P}) + \sum_{i=1}^{n} d_F(\tau_i, \widehat{\tau}_i) \\
&= \sum_{i=1}^{n} \min_{\varsigma\in C} d_F(\widehat{\tau}_i, \varsigma) + \sum_{i=1}^{n} d_F(\tau_i, \widehat{\tau}_i)
\end{aligned}$$

$$\begin{aligned} D_1 &\geq \sum_{i=1}^{n} \min_{\varsigma \in C}(d_F(\widehat{\tau}_i, \varsigma) + d_F(\tau_i, \widehat{\tau}_i)) \\ &\geq \sum_{i=1}^{n} \min_{\varsigma \in C} d_F(\tau_i, \varsigma) \\ &\geq \text{cost}_1(C, P). \end{aligned}$$

Let δ^* be the optimal cost for a solution to the (k, ℓ)-median problem for $P = \{\tau_1, \ldots, \tau_n\}$, and let one such optimal solution be C^*. Analogously to the (k, ℓ)-center problem, it holds that

$$\sum_{i=1}^{n} d_F(\tau_i, \widehat{\tau}_i) \leq \alpha\delta^* \quad \text{and} \quad D_1 \leq \text{cost}_1(C, \widehat{P}) + \alpha\delta^*.$$

An optimal solution C^* is related to D_1 as follows. Let $\omega_i \in C^*$ be the center of this optimal solution which is closest to τ_i, for $1 \leq i \leq n$. We have then

$$\begin{aligned} \delta^* &= \sum_{i=1}^{n} d_F(\tau_i, \omega_i) \\ &\geq \sum_{i=1}^{n} d_F(\widehat{\tau}_i, \omega_i) - \sum_{i=1}^{n} d_F(\tau_i, \widehat{\tau}_i) \\ &\geq \text{cost}_1(C^*, \widehat{P}) - \alpha\delta^* \\ &\geq \frac{1}{\beta}\Big(\text{cost}_1(C, \widehat{P})\Big) - \alpha\delta^* \\ &\geq \frac{1}{\beta}(D_1 - \alpha\delta^*) - \alpha\delta^*. \end{aligned}$$

We conclude that $\delta^* \leq \text{cost}_1(C, P) \leq D_1 \leq (\alpha + \beta + \alpha\beta)\delta^*$, as claimed. □

Theorem 4.7 and Theorem 4.8 claim the quality and the running times of our constant-factor approximation algorithms for (k, ℓ)-clustering problems under the continuous Fréchet distance.

Theorem 4.7. *Given a set of n curves $P = \{\tau_1, \ldots, \tau_n\} \subseteq \Delta_m$ and parameters $k, \ell \in \mathbb{N}$, Algorithm 2 computes an 8-approximation to $\text{opt}_{k,\ell}^{(\infty)}(P)$ under the continuous Fréchet distance and a witness solution in time $O(mnk\ell \log(m\ell))$.*

Proof. The theorem follows from Lemma 4.6 by setting $\alpha = \beta = 2$. We use Lemma 3.41 to compute a 2-approximate ℓ-simplification for each curve (in time $O(m \log m)$ per curve).

Then, we use Gonzales' algorithm which by Theorem 4.4 yields a 2-approximation k-center clustering of the ℓ-simplifications of the curves from P. Each distance computation in Gonzales' algorithm between curves from Δ_ℓ takes $O\left(\ell^2 \log \ell\right)$ time using Alt and Godau's algorithm (Theorem 2.32). To compute the approximate cost, the distance computations between the cluster centers (that are in Δ_ℓ) and the input curves (that are in Δ_m) take $O\left(m\ell \log(m\ell)\right)$ time each. Therefore, the running time of Algorithm 2 is $O\left(mn \log m + nk\ell^2 \log \ell + mnk\ell \log(m\ell)\right) = O\left(mnk\ell \log(m\ell)\right)$. □

If we would first run the Gonzales' algorithm on the input curves, followed by the ℓ-simplification of the obtained cluster centers, the cost of one distance computation would increase to $O\left(m^2 \log m\right)$ and the total running time would be $O\left(m^2 n \log m\right)$.

Theorem 4.8. *Given a set of n curves $P = \{\tau_1, \ldots, \tau_n\} \subseteq \Delta_m$ and parameters $k, \ell \in \mathbb{N}$, Algorithm 2 computes a 68-approximation to $\mathrm{opt}^{(1)}_{k,\ell}(P)$ under the continuous Fréchet distance and a witness solution in time $O\left((nk + k^7 \log^5 n) \cdot m\ell \log(m\ell)\right)$.*

Proof. The theorem follows from Lemma 4.6 by setting $\alpha = 2$ and $\beta = 22$. We use Lemma 3.41 to obtain a 2-approximate ℓ-simplification for each curve in time $O\left(mn \log m\right)$. Then, we use the algorithm of Chen (Theorem 4.5) to obtain the $(10 + \varepsilon)$-approximation of the k-median problem on the ℓ-simplifications, with $0 < \varepsilon < 1$. Thus Chen's algorithm yields an 11-approximation to the discrete problem (where the clusters' centers are constrained to the input set). As we cluster the curves from Δ_ℓ, each distance computation within Chen's algorithm takes $O\left(\ell^2 \log \ell\right) = O\left(1\right)$ time using Alt and Godau's algorithm (Theorem 2.32). Since the Fréchet distance satisfies the triangle inequality, the cluster centers from Chen's algorithm are 2-approximation to the unconstrained case cluster centers, thus $\beta = 22$ is a correct approximation bound. The distance computations between two curves while reporting D_1 takes $O\left(m\ell \log(m\ell)\right)$ each. Therefore, the running time of Algorithm 2 is $O\left(mn \log m + (nk + k^7 \log^5 n) \cdot \ell^2 \log \ell + mnk\ell \log(m\ell)\right) = O\left((nk + k^7 \log^5 n) \cdot m\ell \log(m\ell)\right)$. □

As in the (k, ℓ)-center case, it would be more time consuming to first cluster the original input curves from P, and then to simplify the result, since the distance computation costs $O\left(m^2 \log m\right)$ time each.

It remains to be discussed the differences for (k, ℓ)-clustering under the discrete Fréchet distance. The analysis of Lemma 4.6 remains valid, up to the facts that the distance computation between two curves is now done by the algorithm of Eiter and Mannila (Theorem 2.33), and that we need a minimum-error ℓ-simplification or its α-approximation under the discrete Fréchet distance, that can be plugged in Algorithm 2 instead of Lemma 3.41.

For that we can use the result of Bereg *et al.* [26], that we state as the following lemma adapted here to our notation, and to the one-dimensional curves. The result of Bereg *et al.* holds for curves in $\mathbb{R}^d$, for any dimension $d \in \mathbb{N}$.

Lemma 4.9 (cf. [26] Theorem 3)**.** *Given a curve $\tau : [0, 1] \to \mathbb{R}^d$ with m vertices in general position (for any dimension $d \in \mathbb{N}$), and given a parameter $\ell \in \mathbb{N}$, we can compute in $O(m\ell \log m \log(m/\ell))$ a minimum-error ℓ-simplification π of τ, under the discrete Fréchet distance.*

Using Lemma 4.9 in the proof of Theorem 4.7 we have $\alpha = 1$ and $\beta = 2$. The running time of the distance computations within Gonzales' algorithm is $O\left(\ell^2\right)$ using Eiter and Mannila algorithm. In the computation of the approximation cost the distances are computed in $O(m\ell)$ each. The total running time is thus $O\left(m\ell \log m \log(m/\ell) + nk\ell^2 + mnk\ell\right) = O\left(mnkl + m\ell \log^2 m\right)$. This yields the following theorem.

Theorem 4.10. *Given a set of n curves $P = \{\tau_1, \ldots, \tau_n\} \subseteq \Delta_m$ and parameters $k, \ell \in \mathbb{N}$, Algorithm 2 computes a 5-approximation to $\mathrm{opt}_{k,\ell}^{(\infty)}(P)$ under the discrete Fréchet distance and a witness solution in time $O\left(mnkl + m\ell \log^2 m\right)$.*

For the (k, ℓ)-median clustering the said adaptation of Theorem 4.8 yields a 45-approximation. The running time becomes $O\left(m\ell \log m \log(m/\ell) + (nk + k^7 \log^5 n) \cdot \ell^2 + mnk\ell\right)$. We get the following theorem.

Theorem 4.11. *Given a set of n curves $P = \{\tau_1, \ldots, \tau_n\} \subseteq \Delta_m$ and parameters $k, \ell \in \mathbb{N}$, Algorithm 2 computes a 45-approximation to $\mathrm{opt}_{k,\ell}^{(1)}(P)$ under the discrete Fréchet distance and a witness solution in time $O\left(m\ell \log^2 m + (nk + k^7 \log^5 n) \cdot \ell^2 + mnk\ell\right)$.*

4.4 (k, ℓ)-center $(1 + \varepsilon)$-approximation

In order to improve from a constant-factor approximation discussed in Section 4.3 to a $(1 + \varepsilon)$-approximative solution to the (k, ℓ)-center problem, a common approach is to perform a binary search procedure with approximative cost as a parameter. For this we need an efficient way to generate the candidate solutions for cluster centers in Δ_ℓ that correspond to an approximation of the clustering cost, and that can be evaluated efficiently.

The number of the candidate solutions should not depend on the input size. This is provided by Algorithm 3, where we either decide that we cannot find adequate candidates for cluster centers (i.e. we need to adapt our parameters), or produce a set of candidate solutions of a constant size, where this constant depends on k, ℓ, and ε only. Algorithm 3 uses two positive real parameters α and β. The parameter α approximates the value of

the cost of the optimal solution of (k, ℓ)-center clustering. The parameter β will be used to discretize the candidate set, and it will be related to α.

Algorithm 3: Generate candidates for (k, ℓ)-center from signature vertices

Data: curves $P = \{\tau_1, \ldots, \tau_n\} \subset \Delta_m$, parameters $\alpha, \beta > 0$, $k, \ell \in \mathbb{N}$
Result: candidate set $\Gamma^{k,\ell}_{\alpha,\beta}(P) \subseteq \Delta_\ell$

For each τ_i, let $\mathcal{V}_i$ be the vertex set of its α-signature computed by Algorithm 1
Compute the union U of the intervals $[v - 4\alpha, v + 4\alpha]$ for $v \in \mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i$
if $\mu(U) > 24\alpha k\ell$ **then**
    return the empty set
else
    Discretize U with resolution β, thereby generating a set of vertices $\widehat{\mathcal{V}}$
    return all possible curves consisting of ℓ vertices from $\widehat{\mathcal{V}}$, and denote this set $\Gamma^{k,\ell}_{\alpha,\beta}(P)$

Lemma 4.12. *Given are a set of curves $P = \{\tau_1, \ldots, \tau_n\}$ and parameters $\alpha, \beta > 0$, $k, \ell \in \mathbb{N}$. If $\alpha < \mathrm{opt}^{(\infty)}_{k,\ell}(P)$ then Algorithm 3 concludes correctly that no solution can be found. Otherwise (if $\alpha \geq \mathrm{opt}^{(\infty)}_{k,\ell}(P)$) Algorithm 3 generates a set of candidate solutions $\Gamma^{k,\ell}_{\alpha,\beta}(P) \subseteq \Delta_\ell$ of size at most $(\lfloor 24\alpha k\ell/\beta \rfloor + 6k\ell)^\ell$. The set $\Gamma^{k,\ell}_{\alpha,\beta}(P)$ contains k candidates $\tilde{C} = \{\tilde{\varsigma}_1, \ldots, \tilde{\varsigma}_k\}$ that satisfy*

$$\mathrm{cost}_\infty(P, \tilde{C}) \leq \alpha + \beta.$$

Proof. Let $C = \{\varsigma_1, \ldots, \varsigma_k\}$ denote an optimal solution of (k, ℓ)-center clustering problem for P with cost $\mathrm{opt}^{(\infty)}_{k,\ell}(P)$, and let $\varsigma_i = z_{i1}, \ldots, z_{i\ell}$ denote the vertices for each cluster center $\varsigma_i \in \Delta_\ell$. Consider the union of intervals

$$R = \bigcup_{i=1}^{k} \bigcup_{j=1}^{\ell} [z_{ij} - 4\alpha, z_{ij} + 4\alpha].$$

Let us assume that $\alpha \geq \mathrm{opt}^{(\infty)}_{k,\ell}(P)$. Let $\mathcal{V}_i$ be the vertex set of the α-signature of τ_i computed by Algorithm 1, for $1 \leq i \leq n$, and let $\mathcal{V} = \bigcup_{i=1}^{n} \mathcal{V}_i$. Since for each curve τ_i there is a curve $\varsigma_x \in C$, such that $d_F(\tau_i, \varsigma_x) \leq \alpha$, then by Lemma 3.5 there is a vertex of ς_x at distance at most α to each of the vertices of $\mathcal{V}_i$. Thus all vertices of $\mathcal{V}$ are contained in R.

What can be said for the dual statement, i.e. whether the vertices z_{ij} of the optimal solution C are contained in the set U, where $U = \cup_{v \in \mathcal{V}} [v - 4\alpha, v + 4\alpha]$? Let $\tau_x \in P$ be the curve from the input which is assigned to the center ς_i in the optimal solution C, such that it maximizes the distance $d_F(\tau_x, \varsigma_i)$. If there exists a vertex z_{ij} of the center curve ς_i, which is not contained in U, then let ς_i' be the curve obtained from ς_i by omitting z_{ij}. Let

C' be obtained from C by replacing ς_i with ς_i'. From $d_F(\tau_x, \varsigma_i) \leq \mathrm{opt}_{k,\ell}^{(\infty)}(P) \leq \alpha$ it follows by Theorem 3.8 that $d_F(\tau_x, \varsigma_i') \leq \alpha$. Furthermore it is $\mathrm{cost}_\infty(P, C') \leq \alpha$.

Therefore, let $\widehat{C} = \{\widehat{\varsigma_1}, \ldots, \widehat{\varsigma_k}\}$ denote the solution where all vertices of the cluster centers in C that lie outside U have been omitted. Clearly, U contains all remaining vertices of cluster centers in $\widehat{C}$, and the cost of the (k,ℓ)-center clustering of P with centers $\widehat{C}$ will not increase beyond α, i.e. $\mathrm{cost}_\infty(P, \widehat{C}) \leq \alpha$.

If we discretize U with resolution β, we get the set $\widehat{\mathcal{V}}$. Let the set $\Gamma_{\alpha,\beta}^{k,\ell}(P)$ consist of all curves consisting of ℓ vertices of $\widehat{\mathcal{V}}$. It must contain k candidates $\tilde{C} = \{\tilde{\varsigma_1}, \ldots, \tilde{\varsigma_k}\}$, such that $d_F(\widehat{\varsigma_i}, \tilde{\varsigma_i}) \leq \beta$, for all $1 \leq i \leq k$, and such that

$$\mathrm{cost}_\infty(P, \tilde{C}) \leq \alpha + \beta.$$

Note that R consists of at most $k\ell$ intervals and has measure at most $\mu(R) = 8\alpha k\ell$. The measure of U is bounded by the sum of the measure of R (which covers the intervals centered at signature vertices of $\widehat{\mathcal{V}}$) and the measure of the intervals centered at signature vertices of $\mathcal{V} \setminus \widehat{\mathcal{V}}$. The latter intervals are, in the worst case, centered at each boundary point of R. Thus $\mu(U) \leq \mu(R) + (2k\ell)8\alpha = 24\alpha k\ell$. Furthermore, U consists of at most $\lceil \mu(U)/(8\alpha) \rceil \leq 3k\ell$ intervals, since each interval has measure at least 8α. We conclude that in $\widehat{\mathcal{V}}$ there can be at most $\lfloor 24\alpha k\ell/\beta \rfloor + 6k\ell$ vertices, where the summand $6k\ell$ represents the endpoints of intervals of U. Thus the size of $\Gamma_{\alpha,\beta}^{k,\ell}(P)$ is as claimed.

Therefore, we have the implication $\alpha \geq \mathrm{opt}_{k,\ell}^{(\infty)}(P) \Rightarrow \mu(U) \leq 24\alpha k\ell$. If Algorithm 3 computes that $\mu(U) > 24\alpha k\ell$, this implies that $\alpha < \mathrm{opt}_{k,\ell}^{(\infty)}(P)$ and we conclude correctly that with given α and β no candidate solution set can be found. Otherwise we have correctly generated a candidate set $\Gamma_{\alpha,\beta}^{k,\ell}(P)$ which contains k candidates $\tilde{C}$ that yield the given bound on $\mathrm{cost}_\infty(P, \tilde{C})$. □

Now, the constant-factor approximation algorithm of Section 4.3 gives us bounds for the choice of the parameter α, that we pass to Algorithm 3. We obtain the following theorem.

Theorem 4.13. *Let $0 < \varepsilon < 1$ and $k, \ell \in \mathbb{N}$ be given constants. Given a set of curves $P = \{\tau_1, \ldots, \tau_n\} \subset \Delta_m$, we can compute a $(1+\varepsilon)$-approximation to $\mathrm{opt}_{k,\ell}^{(\infty)}(P)$ and a witness solution in time $O(mn \log m)$.*

Proof. We first apply Algorithm 2 (described in Section 4.3) for (k,ℓ)-center clustering to the input set P. We obtain a solution C with cost D_∞, such that by Theorem 4.7 we can bound the optimal clustering cost by

$$\delta_{\min} = \frac{D_\infty}{8} \leq \mathrm{opt}_{k,\ell}^{(\infty)}(P) \leq D_\infty = \delta_{\max}.$$

According to Lemma 4.12, by running Algorithm 3 with parameters α and β we either correctly decide that the optimal solution cost is larger than α, or produce cluster centers that $(\alpha+\beta)$-approximate the optimal solution. We can apply a binary search procedure on the interval $[\delta_{\min}, \delta_{\max}]$ to find a $(1+\varepsilon)$-approximation solution as follows. In each recursive step of the binary search we set α to be the middle of the current interval, i.e. initially $\alpha = (\delta_{\max} + \delta_{\min})/2$. We run Algorithm 3 on P twice with parameters:

- (α_1, β_1) for the first run, where $\alpha_1 = \alpha/(1+\varepsilon/2)$, $\beta_1 = \alpha_1\varepsilon/2$, and $\alpha_1 + \beta_1 = \alpha$;
- (α_2, β_2) for the second run, where $\alpha_2 = \alpha$, $\beta_2 = \alpha_2\varepsilon/3$, and $\alpha_2 + \beta_2 = \alpha \cdot (1+\varepsilon/3)$.

Thus by Lemma 4.12 we obtain the following knowledge, depending on the outcome of the two runs:

i) First run returns a solution, second produces no solution – this outcome is not possible, as it would imply $\alpha = \alpha_2 < \mathrm{opt}_{k,\ell}^{(\infty)}(P) \leq \alpha$, a contradiction.
ii) None of two runs produces a solution – this implies that $\alpha < \mathrm{opt}_{k,\ell}^{(\infty)}(P)$, thus we rerun the binary search on the interval $[\alpha, \delta_{\max}]$.
iii) Both runs return a solution – thus there is a solution with clustering cost at most α, and we rerun the binary search on the interval $[\delta_{\min}, \alpha]$.
iv) First run produces no solution, but second returns a solution – we have a solution $\tilde{C}$ (from the second run) with the clustering cost satisfying $\alpha/(1+\varepsilon/2) < \mathrm{opt}_{k,\ell}^{(\infty)}(P) \leq \mathrm{cost}_\infty(P, \tilde{C}) \leq \alpha \cdot (1+\varepsilon/3)$. Thus the solution $\tilde{C}$ has the approximation factor $(1+\varepsilon/2)\cdot(1+\varepsilon/3) \leq 1+\varepsilon$ to the optimal solution, since $0 < \varepsilon < 1$.

The number of the binary search steps is at most $O(\log((\delta_{\max} - \delta_{\min})/\varepsilon)) = O(1)$, and each step consists of two runs of Algorithm 3, where the parameters α and β satisfy $\alpha/\beta = O(1/\varepsilon) = O(1)$.

One execution of Algorithm 3 by Lemma 4.12 takes $O(mn)$ time for computing the n signatures (using Algorithm 1), $O(n)$ time to compute U, and $O(1/\beta) = O(1/\varepsilon) = O(1)$ time to generate $\widehat{\mathcal{V}}$. The candidate set $\Gamma_{\alpha,\beta}^{k,\ell}(P)$ has size $(\lfloor 24\alpha k\ell/\beta \rfloor + 6k\ell)^\ell = O\left((k\ell/\varepsilon)^\ell\right) = O(1)$. To evaluate the candidate set it takes to test $\binom{O\left((k\ell/\varepsilon)^\ell\right)}{k} = O(1)$ k-tuples of centers from the candidate set. Evaluating one candidate solution takes kn Fréchet distance computations, where one Fréchet distance computation takes time $O(\ell m \log(\ell m)) = O(m \log m)$ using the algorithm by Alt and Godau (Theorem 2.32). Therefore the total running time is $O(mn \log m)$, as claimed. □

Note that the running time constants that are hidden in the O-notation in Theorem 4.13 are exponential in both input constants k and ℓ.

4.5 (k, ℓ)-median $(1 + \varepsilon)$-approximation

The known $(1+\varepsilon)$-approximation algorithm for k-median clustering problem by Ackermann, Blömer and Sohler [3] for a space with bounded doubling dimension extends into a space $\mathcal{X}$ with unbounded doubling dimension, but that is equipped with arbitrary dissimilarity measure $\mathbf{d}$, such that the sampling property is satisfied, while keeping the asymptotic running time the same (cf. [3]). The adapted result roughly says that we can obtain an efficient $(1+\varepsilon)$-approximation algorithm for the k-median problem on input $P \subseteq \mathcal{X}$, if there is an algorithm that, given a random sample of constant size, returns a set of candidates for the 1-median that contains a $(1 + \varepsilon)$-approximation to the 1-median with constant probability (over the choice of the sample).

We start with restating the sampling property by the following theorem. It was defined by Ackermann, Blömer and Sohler ([3], Property 4.1).

Theorem 4.14 (**Sampling property**). *We say that a dissimilarity measure* $\mathbf{d}$ *on the set* $\mathcal{X}$ *satisfies the (weak)* $[\varepsilon, \gamma]$*-sampling property if and only if there exist integer constants* $m_{\varepsilon,\gamma}$ *and* $t_{\varepsilon,\gamma}$ *such that for each* $P \subseteq \mathcal{X}$ *of size* n *and for each uniform sample multiset* $S \subseteq P$ *of size* $m_{\varepsilon,\gamma}$*, a set* $\Gamma(S) \subseteq \mathcal{X}$ *of size at most* $t_{\varepsilon,\gamma}$ *can be computed satisfying*

$$\mathbf{Pr}\left[(\exists \tilde{c} \in \Gamma(S))\ \sum_{p \in P} \mathbf{d}(p, \tilde{c}) \leq (1 + \varepsilon)\, \mathrm{opt}_1^{(1)}(P)\right] \geq 1 - \gamma.$$

Furthermore, $\Gamma(S)$ *can be computed in time depending on* ε, γ*, and* $m_{\varepsilon,\gamma}$ *only.*

It is likely that the sampling property (Theorem 4.14) does not hold for the Fréchet distance on the set of the one-dimensional curves of complexity at most m, for arbitrary value of m. Furthermore, there is no claim given on the complexity of the candidate center curves. In particular, we need to guarantee that the candidate set $\Gamma(S)$ contains only curves from Δ_ℓ. We will therefore prove a modified sampling property, which allows the size of the sample to depend on ℓ. For that sake, we will first discuss the impact of the input curves on the choice of a candidate center.

4.5.1 Reducing candidate solution set

The following lemma intuitively says that the curves lying far away from a candidate median have little influence on the clustering cost with respect to the candidate median, and thus little influence on the shape of the candidate median.

Lemma 4.15. *Given are a set of n curves $P = \{\tau_1, \ldots, \tau_n\}$ and a curve π. Let R_S be the union of intervals*

$$R_S = \bigcup_{\{\tau_i \in P:\, x_i \leq \frac{2x_1}{\varepsilon}\}} \bigcup_{v \in \mathcal{V}(\sigma_i)} [v - 4x_i, v + 4x_i],$$

where τ_i are sorted in increasing order of $x_i = d_F(\tau_i, \pi)$ and where σ_i is the x_i-signature of $\tau_i \in P$. Then for the curve $\widehat{\pi}$, obtained from π by omitting any subset of vertices lying outside of R_S, it holds that

$$\operatorname{cost}_1(P, \widehat{\pi}) \leq (1 + \varepsilon) \operatorname{cost}_1(P, \pi).$$

Proof. We assume that $\varepsilon \in (0, 2]$. Otherwise the set R_S is empty, and the claim is obvious, since for $\varepsilon > 2$ it holds that $\frac{2x_1}{\varepsilon} < x_1 \leq x_i$, for $1 \leq i \leq n$.

By Lemma 3.5, each curve at distance at most x_i to τ_i has a vertex in each range $[v - x_i, v + x_i]$ centered at vertices $v \in \mathcal{V}(\sigma_i)$. For each signature σ_i, $1 \leq i \leq n$, having at least two vertices, it holds that the curve π has vertices in at least two such ranges, and these vertices will not be omitted. Thus the curve $\widehat{\pi}$ is well defined.

We distinguish two subsets of the curves in P: those τ_i that lie close to π (i.e. those that satisfy $x_i \leq \frac{2x_1}{\varepsilon}$), and the far ones (satisfying $x_i > \frac{2x_1}{\varepsilon}$). For the close curves, it holds by Theorem 3.8 that $d_F(\widehat{\pi}, \tau_i) \leq x_i$.

We now argue that for the curves lying farther away from π the distances to π will increase by a factor of at most $(1 + \varepsilon)$ if we omit the "far" points (and observe $\widehat{\pi}$ instead π). Consider any index i, such that $x_i > \frac{2x_1}{\varepsilon} = \widehat{x}$. By the triangle inequality, it holds that

$$\begin{aligned} d_F(\widehat{\pi}, \tau_i) &\leq d_F(\widehat{\pi}, \tau_1) + d_F(\tau_1, \tau_i) \\ &\leq d_F(\widehat{\pi}, \tau_1) + d_F(\tau_1, \pi) + d_F(\pi, \tau_i) \\ &\leq x_1 + x_1 + x_i \\ &< 2 \cdot \frac{\varepsilon}{2} \cdot x_i + x_i = (1 + \varepsilon) x_i. \end{aligned}$$

Therefore,

$$\begin{aligned} \operatorname{cost}_1(P, \widehat{\pi}) &\leq \sum_{x_i \leq \widehat{x}} d_F(\widehat{\pi}, \tau_i) + \sum_{x_i > \widehat{x}} d_F(\widehat{\pi}, \tau_i) \\ &\leq \sum_{x_i \leq \widehat{x}} d_F(\pi, \tau_i) + \sum_{x_i > \widehat{x}} (1 + \varepsilon) d_F(\pi, \tau_i) \leq (1 + \varepsilon) \operatorname{cost}_1(P, \pi), \end{aligned}$$

as claimed. □

Since computing R_S can depend linearly on the number of input curves n, we want to reduce this dependency, while maintaining the approximation quality. We prove that the basic shape of a candidate median can be approximated based on a constant-size sample. This is stated by the following lemma.

Lemma 4.16. *Given are a set of n curves $P = \{\tau_1, \ldots, \tau_n\}$ and a curve π. There exists an integer constant $m_{\varepsilon,\gamma,\ell} \geq \left\lceil \frac{3\ell}{\varepsilon} \left(\ln \frac{1}{\gamma} + \ln \ell \right) \right\rceil$, such that for each uniform sample multiset $S \subseteq P$ of size $m_{\varepsilon,\gamma,\ell}$, and for a curve $\widehat{\pi}$ obtained from π by omitting any subset of vertices lying outside the union of intervals R_S, defined as:*

$$R_S = \bigcup_{\tau_i \in S} \bigcup_{v \in \mathcal{V}(\sigma_i)} [v - 4x_i, v + 4x_i],$$

where the τ_i are sorted in increasing order of $x_i = d_F(\tau_i, \pi)$ and where σ_i is the x_i-signature of $\tau_i \in S$, it holds that

$$\mathbf{Pr}\left[\mathrm{cost}_1(P, \widehat{\pi}) \leq (1+\varepsilon)\,\mathrm{cost}_1(P, \pi)\right] \geq 1 - \gamma.$$

Proof. If all vertices of π are contained in R_S, then $\pi = \widehat{\pi}$ and the claim is implied. However, this is not necessarily the case. Let $\pi = u_1, \ldots, u_\ell$. In the following, we consider a fixed vertex u_j, $1 \leq j \leq \ell$, and we prove that it is either contained in R_S with a sufficiently high probability, or the cost of a solution will not be increased significantly if we ignore it.

For this purpose, let $T_j \subseteq P$ be the subset of curves τ_i with

$$u_j \in \bigcup_{v \in \mathcal{V}(\sigma_i)} [v - 4x_i, v + 4x_i].$$

If any curve of T_j is contained in our sample S, then u_j is contained in R_S.

We distinguish two cases. If T_j is large enough then u_j is contained in R_S with high probability, otherwise, if T_j is not so large, then we argue that the total change in clustering cost resulting from omitting u_j from π will be insignificant.

Case 1: $|T_j| > \frac{\varepsilon n}{4\ell}$ (T_j is large)

For $|T_j| > \frac{\varepsilon n}{4\ell}$, $1 \leq j \leq \ell$, we want to choose the sample size $m_{\varepsilon,\gamma,\ell}$ such that at least one element of T_j is contained in the sample S. For a fixed index j, it holds that

$$\mathbf{Pr}\left[T_j \cap S = \emptyset\right] \leq \left(1 - \frac{|T_j|}{n}\right)^{m_{\varepsilon,\gamma,\ell}} \leq \left(1 - \frac{\varepsilon}{4\ell}\right)^{m_{\varepsilon,\gamma,\ell}}.$$

We use the union bound inequality to estimate the probability that this event fails for at least one of the sets T_j in question. We choose the parameter $m_{\varepsilon,\gamma,\ell}$ large enough, such that it holds for the failure probability γ that

$$\mathbf{Pr}\left[\left(\bigcup_{j=1}^{\ell} T_j\right) \cap S = \emptyset\right] \leq \sum_{j=1}^{\ell} \mathbf{Pr}\left[T_j \cap S = \emptyset\right] \leq \ell\left(1 - \frac{\varepsilon}{4\ell}\right)^{m_{\varepsilon,\gamma,\ell}} \leq \gamma.$$

The last inequality can be transformed into

$$m_{\varepsilon,\gamma,\ell} \geq \frac{\ln\frac{\ell}{\gamma}}{\ln\left(1+\frac{\varepsilon}{4\ell-\varepsilon}\right)} \geq \frac{4\ell-\varepsilon}{\varepsilon}\ln\frac{\ell}{\gamma} \geq \frac{3\ell}{\varepsilon}\ln\frac{\ell}{\gamma},$$

using inequality $\ln(1+x) \leq x$, that holds for all $x \geq 0$, and since $\ell \geq 2$ and $\varepsilon \in [0,2)$. Thus it suffices to choose

$$m_{\varepsilon,\gamma,\ell} \geq \left\lceil \frac{3\ell}{\varepsilon}\left(\ln\frac{1}{\gamma} + \ln\ell\right)\right\rceil$$

to obtain that, with a probability at least $1-\gamma$ for all $1 \leq j \leq \ell$ simultaneously, we have at least one element of T_j in S, if $|T_j| \geq \frac{\varepsilon n}{4\ell}$.

Case 2: $|T_j| \leq \frac{\varepsilon n}{4\ell}$ (T_j is small)

Consider the set of curves $\mathcal{T} = \{T_j \colon 1 \leq j \leq \ell, T_j \cap S = \emptyset\}$. Following the analysis of the first case it holds that

$$\mathbf{Pr}\left[\mathcal{T} \not\subseteq \{T_j \colon |T_j| \leq \frac{\varepsilon n}{4\ell}\}\right] = \mathbf{Pr}\left[(\exists T_j \in \mathcal{T})\,|T_j| > \frac{\varepsilon n}{4\ell}\right] \leq \gamma.$$

Then we have that with probability at least $1-\gamma$ is $\mathcal{T} \subseteq \{T_j \colon 1 \leq j \leq \ell, |T_j| \leq \frac{\varepsilon n}{4\ell}\}$. For the rest of the case analysis we assume that this event happens.

Let $\tilde{\pi}$ denote the curve obtained from π by removing all vertices lying outside R_S. This is equivalent to removing all vertices u_j with $T_j \in \mathcal{T}$. Namely, if for fixed u_j it is $(\forall \tau_i \in S)\, u_j \notin \cup_{v\in\mathcal{V}(\sigma_i)}[v-4x_i, v+4x_i]$, then $(\forall \tau_i \in S)\, \tau_i \notin T_j \Leftrightarrow T_j \cap S = \emptyset \Leftrightarrow T_j \in \mathcal{T}$.

In the following, let $P_{\mathcal{T}} = \bigcup_{T'\in\mathcal{T}} T'$ be the set of input curves that are contained in one of the sets in $\mathcal{T}$. For any curve $\tau_i \in P \setminus P_{\mathcal{T}}$ it holds that $\tau_i \notin T_j$ for any $T_j \in \mathcal{T}$. This implies that none of the vertices u_j of π is contained in the union of the ranges around the vertices of the signature σ_i, thus by Theorem 3.8, we can exclude u_j from π without increasing the distance of the remaining curve to τ_i beyond x_i. Therefore, for such a curve $\tau_i \in P \setminus P_{\mathcal{T}}$ it holds that $d_F(\tilde{\pi}, \tau_i) \leq x_i = d_F(\pi, \tau_i)$.

Let τ_q be the curve in $P \setminus P_{\mathcal{T}}$ with minimal distance to π (i.e. with smallest index q). At least half of the input curves have to lie within a radius of $r = \frac{2}{n}\operatorname{cost}_1(P, \pi)$ from π (twice the average distance of the input curves to π). Otherwise we would have

$$\begin{aligned}\operatorname{cost}_1(P, \pi) &> \sum_{d_F(\tau_i, \pi) \leq r} d_F(\tau_i, \pi) + \lceil \frac{n}{2} \rceil \cdot \frac{2}{n} \operatorname{cost}_1(P, \pi) \\ &\geq \sum_{d_F(\tau_i, \pi) \leq r} d_F(\tau_i, \pi) + \operatorname{cost}_1(P, \pi),\end{aligned}$$

a contradiction. Furthermore, the set $P_{\mathcal{T}}$ has size less than $n/2$ (with probability at least $1 - \gamma$), since $\mathcal{T} \subseteq \{T_j \colon |T_j| \leq \frac{\varepsilon n}{4\ell}\}$, and there are at most ℓ "small" sets, thus $|P_{\mathcal{T}}| \leq \ell \cdot \frac{\varepsilon n}{4\ell} < \frac{n}{2}$. Therefore with probability at least $1 - \gamma$ we conclude that $x_q \leq \frac{2}{n}\operatorname{cost}_1(P, \pi)$ (otherwise all elements of $P \setminus P_{\mathcal{T}}$ would lie at distance to π larger than r).

Using triangle inequality we conclude that

$$\begin{aligned}\operatorname{cost}_1(P, \tilde{\pi}) &= \operatorname{cost}_1(P \setminus P_{\mathcal{T}}, \tilde{\pi}) + \operatorname{cost}_1(P_{\mathcal{T}}, \tilde{\pi}) \\ &\leq \sum_{\tau \in P \setminus P_{\mathcal{T}}} d_F(\tilde{\pi}, \tau) + \sum_{\tau \in P_{\mathcal{T}}} (d_F(\tau, \pi) + d_F(\pi, \tau_q) + d_F(\tau_q, \tilde{\pi})) \\ &\leq \sum_{\tau \in P \setminus P_{\mathcal{T}}} d_F(\pi, \tau) + \operatorname{cost}_1(P_{\mathcal{T}}, \pi) + |P_{\mathcal{T}}| \cdot (d_F(\pi, \tau_q) + d_F(\tau_q, \tilde{\pi})) \\ &\leq \operatorname{cost}_1(P \setminus P_{\mathcal{T}}, \pi) + \operatorname{cost}_1(P_{\mathcal{T}}, \pi) + |P_{\mathcal{T}}| \cdot 2x_q \\ &\leq \operatorname{cost}_1(P, \pi) + \frac{\varepsilon n}{4} \cdot \frac{4 \operatorname{cost}_1(P, \pi)}{n} \\ &= (1 + \varepsilon) \operatorname{cost}_1(P, \pi),\end{aligned}$$

with probability at least $1 - \gamma$, as claimed. □

4.5.2 Generating candidate solutions

After we have shown how to approximate the cost function based on a constant-sized sample, we need to generate a set of candidates for the $(1, \ell)$-median clustering center curve, based on the sample set. This is done using an algorithm, that is similar in form to Algorithm 3 for (k, ℓ)-center problem. The roles of the positive real parameters α and β are analogous to those in Algorithm 3, and will both be later related to a constant-factor approximation solution.

Algorithm 4: Generate candidates for $(1, \ell)$-median from signature vertices

Data: curves $S = \{\tau_1, \ldots, \tau_s\} \subset \Delta_m$, parameters $\alpha, \beta > 0$, $\ell \in \mathbb{N}$
Result: candidate set $\Gamma^{1,\ell}_{\alpha,\beta}(S) \subseteq \Delta_\ell$

1 For each τ_i, let $\mathcal{V}_i$ be the vertex set of the signature of size at most $\ell + 3$ (Theorem 3.39)
2 Compute the union U of the intervals $[v - 8\alpha, v + 8\alpha]$ for $v \in \mathcal{V} = \bigcup_{i=1}^{s} \mathcal{V}_i$
3 Discretize U with resolution β, thereby generating a set of vertices $\widehat{\mathcal{V}}$
4 **return** all possible curves consisting of ℓ vertices from $\widehat{\mathcal{V}}$

Algorithm 4 generates a candidate set, whose properties we prove next. The proof of Lemma 4.17 serves as a basis for the proof of the modified sampling property in Theorem 4.20.

Lemma 4.17. *Given are a set of curves $S = \{\tau_1, \ldots, \tau_s\}$ and parameters $\alpha, \beta > 0$, $\varepsilon \in (0, 2]$, and $\ell \in \mathbb{N}$, with $\alpha \geq \min_{i \in \{1,\ldots,s\}} \frac{d_F(\tau_i, \varsigma_s)}{\varepsilon}$, where ς_s denotes an optimal $(1, \ell)$-median clustering center of S. There exists $\widehat{\varsigma} \in \Delta_\ell$ with*

$$\mathrm{cost}_1(S, \widehat{\varsigma}) \leq (1 + \varepsilon) \, \mathrm{opt}^{(1)}_{1,\ell}(S),$$

and Algorithm 4 computes a set of candidates $\Gamma^{1,\ell}_{\alpha,\beta}(S) \subseteq \Delta_\ell$ of size $\left(\frac{16\alpha s(\ell+3)}{\beta}\right)^\ell$ which contains an element $\tilde{\varsigma}$, such that

$$d_F(\widehat{\varsigma}, \tilde{\varsigma}) \leq \beta.$$

Proof. Let $\tau_1, \ldots, \tau_s$ denote the input curves in the increasing order of their distance to ς_s, denoted by $x_i = d_F(\varsigma_s, \tau_i)$. For every τ_i, consider its x_i-signature denoted by σ_i. Using the same arguments as in the proof of Lemma 4.12, by Lemma 3.5, each vertex of ς_s lies within distance $4x_i$ to a vertex of some x_i-signature σ_i otherwise we can omit it by Theorem 3.8. Hence, we can bound our search for candidate vertices of the curve ς_s to the union the intervals.

$$\bigcup_{\tau_i \in S} \bigcup_{v \in \mathcal{V}(\sigma_i)} [v - 4x_i, v + 4x_i].$$

Since x_i could be considerably large, we cannot cover this entire region with candidates. Instead, we consider the following union of intervals:

$$R_S = \bigcup_{\{\tau_i \in S \colon x_i \leq \widehat{x}\}} \bigcup_{v \in \mathcal{V}(\sigma_i)} [v - 4x_i, v + 4x_i],$$

with $\widehat{x} = \frac{2x_1}{\varepsilon}$. Now, let $\widehat{\varsigma}$ be the curve obtained from ς_s by omitting all vertices that do not lie in R_S. Lemma 4.15 implies that $\mathrm{cost}_1(S,\widehat{\varsigma}) \leq (1+\varepsilon)\,\mathrm{cost}_1(S,\varsigma_s)$, and the first claim of the lemma is proven.

Now we can relate the set of the candidate curves returned by Algorithm 4 to the curve $\widehat{c}$ (and implicitely to ς_s) as follows. By Corollary 3.6 we have $\ell \geq |\mathcal{V}(\varsigma_s)\,| \geq |\mathcal{V}(\sigma_i)\,| - 2$, since $\varsigma_s \in \Delta_\ell$ and $d_F(\tau_i,\varsigma_s) = x_i \geq d_F(\tau_i,\sigma_i)$. Thus, it is $|\mathcal{V}(\sigma_i)\,| \leq \ell + 2$, for $1 \leq i \leq s$. If a signature of size $\ell+3$ does not exist, then by the general position assumption, there must be a signature of size $\ell+2$. We conclude that by Lemma 3.34, the vertices of σ_i are contained in the set of signature vertices computed by Algorithm 4 (i.e. of the canonical signatures of size at most $\ell+3$).

Since we have chosen $\alpha \geq \min_{1\leq i\leq s} \frac{d_F(\tau_i,\varsigma_s)}{\varepsilon}$, it is $2\alpha \geq \frac{2x_1}{\varepsilon} = \widehat{x}$. Therefore, for the intervals used in Algorithm 4, it holds that $[v-4x_i, v+4x_i] \subseteq [v-4\widehat{x}, v+4\widehat{x}] \subseteq [v-8\alpha, v+8\alpha]$, with $x_i \leq \widehat{x}$. Hence the union U of the intervals covers the set R_S, and if we discretize it with resolution β, we conclude that such generated candidate set contains a curve $\tilde{\varsigma}$ that lies within Fréchet distance β of $\widehat{\varsigma}$, as claimed. For the measure of the union U it holds that $\mu(U) \leq 16\alpha s(\ell+3)$. This implies the size of the generated set $\Gamma^{1,\ell}_{\alpha,\beta}(S)$. □

Before we prove the modified sampling property we need to prove the following two lemmas. Lemma 4.18 bounds the probability that the cost of $(1,\ell)$-clustering of S with center at the optimal median of P deviates significantly. Lemma 4.19 gives lower bound on the cost of the optimal $(1,\ell)$-median clustering of the sample, related to the optimal $(1,\ell)$-median clustering center of the whole input. The proof technique of this lemma is inspired by a result by Kumar, Sabharwal and Sen [126] (cf. their Theorem 5.4).

Lemma 4.18. *Let $0 < \gamma \leq 1$. Given a set $P = \{\tau_1,\ldots,\tau_n\}$ of curves from Δ_ℓ, for each uniform sample multiset $S \subseteq P$ it holds that*

$$\mathbf{Pr}\left[\mathrm{cost}_1(S,\varsigma) \geq \frac{|S|}{\gamma n}\,\mathrm{opt}^{(1)}_{1,\ell}(P)\right] \leq \gamma,$$

where ς is an optimal $(1,\ell)$-median center of P.

Proof. Let $S = \{\tau'_1,\ldots,\tau'_s\} \subseteq P$. It holds that

$$\mathbb{E}\left[\mathrm{cost}_1(S,\varsigma)\right] = \mathbb{E}\left[\sum_{i=1}^{s} d_F\big(\tau'_i,\varsigma\big)\right] = \sum_{i=1}^{s}\mathbb{E}\left[d_F\big(\tau'_i,\varsigma\big)\right] = |S|\sum_{j=1}^{n}\frac{d_F(\tau_j,\varsigma)}{n} = \frac{|S|}{n}\,\mathrm{opt}^{(1)}_{1,\ell}(P).$$

Since $\mathrm{cost}_1(S,\varsigma)$ is a non-negative random variable, we can apply Markov's inequality and obtain

$$\mathbf{Pr}\left[\mathrm{cost}_1(S,\varsigma) \geq \frac{\mathbb{E}\left[\mathrm{cost}_1(S,\varsigma)\right]}{\gamma}\right] \leq \gamma,$$

which implies the claim. □

Lemma 4.19. *Let $0 < \gamma \leq 1$. Given a set of curves P, for each uniform sample multiset $S \subseteq P$ of size at least $\lceil 6.5 \ln \frac{1}{\gamma} \rceil + 2$ it holds that*

$$\mathbf{Pr}\left[12 \operatorname{opt}^{(1)}_{1,\ell}(S) \geq \min_{\tau \in P} d_F(\tau, \varsigma)\right] \geq 1 - \gamma,$$

where ς denotes an optimal $(1,\ell)$-median clustering center of P.

Proof. We analyze two cases, depending on whether the input curves are concentrated close to a curve in Δ_ℓ or not, in particular, if there exists a curve $\rho \in \Delta_\ell$, such that $|\{\tau \in P \colon d_F(\rho,\tau) \leq r\}| \geq \frac{5}{7}|P|$, where $r = d_F(\rho,\varsigma)/5$.

Case 1: There exists $\rho \in \Delta_\ell$ with $|\{\tau \in P \colon d_F(\rho,\tau) \leq r\}| \geq \frac{5}{7}|P|$.

In this case we assume that a large fraction of P lies within a small ball far away from the optimal center. We let

$$Q = \{\tau \in P \colon d_F(\rho,\tau) < 2r\},$$

and we claim that Q has size at most $\frac{6}{7}|P|$. If we assume the opposite (i.e. $|Q| > \frac{6}{7}|P|$), then it follows by the triangle inequality that

$$\begin{aligned}
\operatorname{cost}_1(P,\varsigma) - \operatorname{cost}_1(P,\rho) &= \sum_{\tau \in Q} (d_F(\tau,\varsigma) - d_F(\tau,\rho)) + \sum_{\tau \in P \setminus Q} (d_F(\tau,\varsigma) - d_F(\tau,\rho)) \\
&\geq \sum_{\tau \in Q} (d_F(\rho,\varsigma) - 2d_F(\tau,\rho)) + \sum_{\tau \in P \setminus Q} (-d_F(\rho,\varsigma)) \\
&> |Q| \cdot \left(d_F(\rho,\varsigma) - \frac{4}{5} d_F(\rho,\varsigma)\right) - |P \setminus Q| \cdot d_F(\rho,\varsigma) \\
&> \frac{6}{7}|P| \cdot r - \frac{5}{7}|P| \cdot r = \frac{1}{7}|P| \cdot r \geq 0.
\end{aligned}$$

This would imply that ς is not optimal, a contradiction.

Now we analyze the event that at least one curve of P lies within Fréchet distance r of ρ and at least one curve lies farther than $2r$ from ρ. If this event happens, then we have that

$$\operatorname{opt}^{(1)}_{1,\ell}(S) \geq \max_{\sigma',\sigma'' \in S} d_F(\sigma',\sigma'') \geq r \geq \frac{\min_{\tau \in P} d_F(\varsigma,\tau)}{6}. \tag{4.3}$$

The first inequality in Equation (4.3) results from the triangle inequality. The second inequality in (4.3) results from the event condition. If the third inequality

would not hold, then there would exist $\tau' \in P$ with $d_F(\rho, \tau') \leq r$, implying that $d_F(\varsigma, \tau') \leq d_F(\varsigma, \rho) + d_F(\rho, \tau') \leq 5r + r = 6r$, a contradiction.

From the assumptions about the size of the sets $\{\tau \in P\colon d_F(\rho, \tau) \leq r\}$ and Q, it follows that for the ith sample curve $\sigma_i \in S$, it is

$$\mathbf{Pr}\left[d_F(\sigma_i, \rho) \leq r\right] \geq \frac{5}{7} \quad \text{and} \quad \mathbf{Pr}\left[d_F(\sigma_i, \rho) \geq 2r\right] \geq \frac{1}{7}.$$

In order to have

$$\mathbf{Pr}\left[(\exists \sigma', \sigma'' \in S)\ d_F\big(\rho, \sigma'\big) \leq r \wedge d_F\big(\rho, \sigma''\big) \geq 2r\right] \geq 1 - \gamma,$$

we observe that, for the complementary event holds

$$\begin{aligned}\mathbf{Pr}\left[(\forall \sigma_i \in S)(d_F(\rho, \sigma_i) > r) \vee (\forall \sigma_i \in S)(d_F(\rho, \sigma_i) < 2r)\right] \leq \\ \leq \left(\frac{2}{7}\right)^{|S|} + \left(\frac{6}{7}\right)^{|S|} \leq \frac{4}{3} \cdot \left(\frac{6}{7}\right)^{|S|} \leq \gamma,\end{aligned}$$

using union bound inequality. It suffices to take $|S| \geq \lceil 6.5 \ln \frac{1}{\gamma} \rceil + 2 \geq (\ln \frac{1}{\gamma} + \ln \frac{4}{3}) / \ln \frac{7}{6}$ samples for the last inequality to hold. Thus, Equation (4.3) (that implies the claim of the lemma) holds with probability of at least $1 - \gamma$.

Case 2: There is no $\rho \in \Delta_\ell$ with $|\{\tau \in P\colon d_F(\rho, \tau) \leq r\}| \geq \frac{5}{7}|P|$.

Let σ_1 be the first sample curve and let $\widehat{\sigma}_1$ be its minimum-error ℓ-simplification (see Definition 3.1). We first note that if $12 d_F(\sigma_1, \widehat{\sigma}_1) \geq d_F(\sigma_1, \varsigma)$, then it holds

$$12 \operatorname{opt}^{(1)}_{1,\ell}(S) \geq 12 d_F(\sigma_1, \varsigma) \geq 12 d_F(\sigma_1, \widehat{\sigma}_1) \geq d_F(\sigma_1, \varsigma) \geq \min_{\tau \in P} d_F(\tau, \varsigma), \tag{4.4}$$

and the claim of the lemma holds with probability 1. For the rest of the case analysis we assume that

$$12 d_F(\sigma_1, \widehat{\sigma}_1) < d_F(\sigma_1, \varsigma). \tag{4.5}$$

The case definition provides

$$|\{\tau \in P\colon d_F(\widehat{\sigma}_1, \tau) \leq r\}| < \frac{5}{7}|P|,$$

for $r = d_F(\widehat{\sigma}_1, \varsigma)/5$. Thus, for each of the remaining sample curves σ_i, for $1 < i \leq |S|$, is

$$\mathbf{Pr}\left[d_F(\widehat{\sigma}_1, \sigma_i) > r\right] \geq \frac{2}{7}. \tag{4.6}$$

It suffices to take $|S| \geq 3 \ln \frac{1}{\gamma} + 1 \geq \ln \frac{1}{\gamma} / \ln \frac{7}{5} + 1$ samples, in order to ensure that there is at least one index i, $1 < i \leq |S|$, such that $d_F(\widehat{\sigma}_1, \sigma_i) > r$, with probability of at least $1 - \gamma$. Let j be one such index, then it holds by the triangle inequality that

$$\begin{aligned} d_F(\sigma_1, \sigma_j) &\geq d_F(\sigma_j, \widehat{\sigma}_1) - d_F(\widehat{\sigma}_1, \sigma_1) > r - d_F(\widehat{\sigma}_1, \sigma_1) = \frac{d_F(\widehat{\sigma}_1, \varsigma)}{5} - d_F(\widehat{\sigma}_1, \sigma_1) \\ &\geq \frac{d_F(\sigma_1, \varsigma) - d_F(\sigma_1, \widehat{\sigma}_1)}{5} - d_F(\widehat{\sigma}_1, \sigma_1) = \frac{d_F(\sigma_1, \varsigma)}{5} - \frac{6 d_F(\sigma_1, \widehat{\sigma}_1)}{5} \\ &\overset{(4.5)}{>} \frac{d_F(\sigma_1, \varsigma)}{10}. \end{aligned}$$

We conclude using triangle inequality, that with probability at least $1 - \gamma$, it is

$$\begin{aligned} 10 \operatorname{opt}^{(1)}_{1,\ell}(S) &\geq 10 \left(d_F(\sigma_1, \varsigma) + d_F(\sigma_j, \varsigma) \right) \geq 10 d_F(\sigma_1, \sigma_j) \\ &> d_F(\sigma_1, \varsigma) \geq \min_{\tau \in P} d_F(\tau, \varsigma), \end{aligned} \tag{4.7}$$

which completes the proof of the lemma. □

4.5.3 Modified sampling property

We are now ready to prove the modified sampling property. Note that in comparison to the sampling property of Ackermann, Blömer and Sohler [3] (cf. Theorem 4.14) the running time to compute a candidate set depends on the additional parameter m, but now we have a guarantee on the complexity of the candidate curves.

Theorem 4.20 (**Modified sampling property**). *Let $\varepsilon \in (0, 2]$ and $\gamma \in (0, 1]$. There exist integer constants $m_{\varepsilon,\gamma,\ell}$ and $t_{\varepsilon,\gamma,\ell}$ such that given a set of curves $P = \{\tau_1, \ldots, \tau_n\}$ from Δ_m for a uniform sample multiset $S \subseteq P$ of size $m_{\varepsilon,\gamma,\ell}$ we can generate a candidate set $\Gamma(S) \subset \Delta_\ell$ of size $t_{\varepsilon,\gamma,\ell}$ satisfying*

$$\mathbf{Pr}\left[(\exists q \in \Gamma(S)) \ \operatorname{cost}_1(P, q) \leq (1 + \varepsilon) \operatorname{opt}^{(1)}_{1,\ell}(P) \right] \geq 1 - \gamma.$$

Furthermore, we can compute $\Gamma(S)$ in time depending only on $\ell, \gamma, \varepsilon$, and m.

Proof. Let $\gamma' = \frac{\gamma}{4}$ and $\varepsilon' = \frac{\varepsilon}{4}$. Let ς denote an optimal $(1, \ell)$-median of P and let ς_s denote an optimal $(1, \ell)$-median of S. We use Algorithm 2 for (k, ℓ)-median problem described in Section 4.3 to compute a constant-factor approximation D_1 to $\operatorname{opt}^{(1)}_{1,\ell}(S)$ and obtain an interval $[\delta_S^{\min}, \delta_S^{\max}]$ which contains $\operatorname{opt}^{(1)}_{1,\ell}(S)$. By Theorem 4.8 it holds that $\delta_S^{\max} = D_1$ and $\delta_S^{\min} = D_1/68$. We apply Algorithm 4 to S with parameters

$$\alpha = \frac{12 \delta_S^{\max}}{\varepsilon'} \quad \text{and} \quad \beta = \frac{\varepsilon' \gamma' \delta_S^{\min}}{|S|},$$

to obtain a set $\Gamma^{1,\ell}_{\alpha,\beta}(S)$. We claim that the set $\Gamma^{1,\ell}_{\alpha,\beta}(S)$ satisfies the properties of $\Gamma(S)$.

As in Lemma 4.17, we argument as follows. Let $\tau_1,\ldots,\tau_n$ denote the input curves in the increasing order of their distance to ς, denoted by $x_i = d_F(\varsigma,\tau_i)$. For every τ_i, consider its x_i-signature denoted by σ_i. By Lemma 3.5, each vertex of ς lies within distance $4x_i$ to a vertex of some signature σ_i, otherwise we can omit it by Theorem 3.8. Hence, there must be a $(1,\ell)$-median curve whose vertex set is contained in the union of the intervals

$$\bigcup_{\tau_i\in P}\bigcup_{v\in\mathcal{V}(\sigma_i)}[v-4x_i, v+4x_i].$$

Let this solution be denoted ς.

We first consider the following union of intervals:

$$R_1 = \bigcup_{\tau_i\in S}\bigcup_{v\in\mathcal{V}(\sigma_i)}[v-4x_i, v+4x_i].$$

Let $\widehat{\varsigma_1}$ be the median curve obtained from ς by omitting all vertices that do not lie in R_1. Lemma 4.16 implies

$$\mathbf{Pr}\left[\mathrm{cost}_1(P,\widehat{\varsigma_1}) \le (1+\varepsilon')\,\mathrm{cost}_1(P,\varsigma)\right] \ge 1-\gamma', \tag{4.8}$$

if we choose $|S| \ge \left\lceil \frac{3\ell}{\varepsilon'}\left(\ln\frac{1}{\gamma'} + \ln\ell\right)\right\rceil$.

Second, we consider the following union of intervals:

$$R_2 = \bigcup_{\{\tau_i\in P:\, x_i\le\widehat{x}\}}\bigcup_{v\in\mathcal{V}(\sigma_i)}[v-4x_i, v+4x_i],$$

where $\widehat{x} = \frac{2x_1}{\varepsilon'}$. Let $\widehat{\varsigma_2}$ be the median curve obtained from $\widehat{\varsigma_1}$ by omitting all vertices that do not lie in R_2. We can apply Lemma 4.15 and obtain

$$\mathrm{cost}_1(P,\widehat{\varsigma_2}) \le (1+\varepsilon')\,\mathrm{cost}_1(P,\widehat{\varsigma_1}). \tag{4.9}$$

From Lemma 4.19 follows that, if we take the set of sample curves $|S| \ge \lceil 6.5\ln\frac{1}{\gamma'}\rceil + 2$, then it holds that $\mathbf{Pr}\left[x_1 \le 12\,\mathrm{opt}^{(1)}_{1,\ell}(S)\right] \ge 1-\gamma'$. Hence with the same probability it holds that

$$8\alpha = 8\cdot\frac{12\delta_S^{\max}}{\varepsilon'} \ge 8\cdot\frac{12\,\mathrm{opt}^{(1)}_{1,\ell}(S)}{\varepsilon'} \ge 8\cdot\frac{x_1}{\varepsilon'} = 4\widehat{x}, \tag{4.10}$$

since we have chosen $\alpha = \frac{12\delta_S^{\max}}{\varepsilon'}$ and $\widehat{x} = \frac{2x_1}{\varepsilon'}$. This relates the intervals of Algorithm 4 and R_2. From Equation (4.10) we have that $\alpha \ge 12\,\mathrm{opt}^{(1)}_{1,\ell}(S)/\varepsilon' \ge 12\min d_F(\tau_i,\varsigma_s)/\varepsilon'$.

Therefore the conditions of Lemma 4.17 are fulfilled and its claims hold with probability at least $1-\gamma'$. In particular, with probability $1-\gamma'$, the generated set $\Gamma^{1,\ell}_{\alpha,\beta}(S)$ contains a curve $\tilde{\varsigma}$ which lies within Fréchet distance β of $\widehat{\varsigma}_2$.

Lemma 4.18 implies that with probability at least $1-\gamma'$ it holds that

$$\operatorname{opt}^{(1)}_{1,\ell}(S) \leq \operatorname{cost}_1(S,\varsigma) \leq \frac{|S|}{\gamma' n} \operatorname{opt}^{(1)}_{1,\ell}(P).$$

Thus, with the same probability it holds that

$$\beta = \frac{\varepsilon'\gamma'\delta_S^{\min}}{|S|} \leq \frac{\varepsilon'\gamma' \operatorname{opt}^{(1)}_{1,\ell}(S)}{|S|} \leq \frac{\varepsilon' \operatorname{opt}^{(1)}_{1,\ell}(P)}{n}. \tag{4.11}$$

Using union bound inequality we conclude that with probability at least $1-3\gamma' > 1-\gamma$ the events of Equations (4.8), (4.10), and (4.11) simultaneously occur, and thus there exists a candidate $\tilde{\varsigma} \in \Gamma^{1,\ell}_{\alpha,\beta}(S)$ such that

$$\begin{aligned}\operatorname{cost}_1(P,\tilde{\varsigma}) &\leq \sum_{\tau\in P} \left(d_F(\tau,\widehat{\varsigma}_2) + d_F(\widehat{\varsigma}_2,\tilde{\varsigma})\right) \leq \operatorname{cost}_1(P,\widehat{\varsigma}_2) + \beta n \overset{(4.9)}{\leq} (1+\varepsilon')\operatorname{cost}_1(P,\widehat{\varsigma}_1) + \beta n \\ &\overset{(4.8)}{\leq} (1+\varepsilon')^2 \operatorname{cost}_1(P,\varsigma) + \beta n \overset{(4.11)}{\leq} ((1+\varepsilon')^2+\varepsilon') \operatorname{opt}^{(1)}_{1,\ell}(P) \leq (1+\varepsilon)\operatorname{opt}^{(1)}_{1,\ell}(P).\end{aligned}$$

The last inequality results from $\varepsilon \in (0,2]$. The size of the sampled multiset is $m_{\varepsilon,\gamma,\ell} = |S| = \lceil \frac{12\ell}{\varepsilon}\left(\ln\frac{4}{\gamma} + \ln\ell\right)\rceil$, since $\lceil \frac{3\ell}{\varepsilon'}\left(\ln\frac{1}{\gamma'} + \ln\ell\right)\rceil \geq \lceil 6.5\ln\frac{1}{\gamma'}\rceil + 2$. Furthermore, by Lemma 4.17 the size of $\Gamma^{1,\ell}_{\alpha,\beta}(S)$ is bounded as follows

$$\begin{aligned}t_{\varepsilon,\gamma,\ell} &\leq \left(\frac{16\alpha|S|(\ell+3)}{\beta}\right)^{\ell} = \left(\left(16\cdot 12\cdot 68\cdot 4^3\right)\frac{|S|(\ell+3)}{\varepsilon^2\gamma}\right)^{\ell} \\ &\leq c_1 \cdot \left(\frac{\ell^3}{\varepsilon^4\gamma}\left(\log^2\frac{1}{\gamma} + \log^2\ell\right)\right)^{\ell},\end{aligned}$$

where c_1 is a sufficiently large constant. The set $\Gamma^{1,\ell}_{\alpha,\beta}(S)$ is computed in time that depends only on ε, γ, ℓ, and m. Namely, we need $O(|S|)$ time for the sampling. For a constant-factor approximation by Algorithm 2 we need $O(|S|m\ell\log(m\ell))$ time by Theorem 4.8. By Theorem 3.39 we compute the signatures in Algorithm 4 in $O(|S|m\log m)$ time. The discretization of the ranges around signature vertices requires $O(|S|/(\varepsilon\gamma))$, and the computing of the candidate curves requires $O(t_{\varepsilon,\gamma,\ell})$ time. If we observe the parameters ε, γ, and ℓ as constants, then the time needed to compute $\Gamma(S)$ is $O(m\log m)$. This completes the proof of the theorem. □

Our definition of the (k,ℓ)-median clustering problem is in the metric space (Δ_m, d_F). However, it corresponds to the classical definition of the k-median problem (cf. Section 2.4), if the ground set is $\mathcal{X} = \Delta_\ell \cup P$, $P \subset \Delta_m$, and the distance measure $\mathbf{d}$ is defined for $x, y \in \mathcal{X}$ as

$$\mathbf{d}(x,y) = \begin{cases} \infty & \text{if } x,y \in P \text{ and } x \neq y, \\ 0 & \text{if } x,y \in P \text{ and } x = y, \\ d_F(x,y) & \text{otherwise.}^{13} \end{cases}$$

Such defined distance measure $\mathbf{d}$ on $\Delta_\ell \cup P$ is not a metric, since it does not satisfy the triangle inequality, i.e. it does not hold $\mathbf{d}(x,y) + \mathbf{d}(y,z) \geq \mathbf{d}(x,z)$ if $x, z \in P$, $x \neq z$, and $y \notin P$. Other properties of Definition 2.1 are clearly satisfied. But the analysis of Ackermann, Blömer and Sohler [3] requires that $\mathbf{d}$ be only a dissimilarity measure, and not necessarily a metric. Therefore, we can use Theorem 4.20 in the space $\Delta_\ell \cup P$, and incorporate it (instead of Theorem 4.14) into the analysis of the k-median clustering algorithm by Ackermann, Blömer and Sohler [3]. Their result is stated as the following theorem.

Theorem 4.21 (cf. [3] Theorem 1.1)**.** *Given are $k \in \mathbb{N}$ and $0 < \varepsilon, \gamma < 1$. Let $\mathcal{X}$ be a ground set with dissimilarity measure $\mathbf{d}$, such that it satisfies the sampling property (Theorem 4.14). Let $m_{\varepsilon,\ell}$ and $t_{\varepsilon,\ell}$ be the constants provided by Theorem 4.14. There exists an algorithm that with constant probability returns a $(1+\varepsilon)$-approximation of the k-median clustering problem with respect to $\mathbf{d}$ for input instance $P \subset \mathcal{X}$, $|P| = n$, and that requires at most*

$$n \cdot 2^{O\left(k m_{\varepsilon/3,\gamma} \log((k/\varepsilon)\cdot m_{\varepsilon/3,\gamma})\right)}$$

arithmetic operations, including evaluations of the clustering cost.

The distance computations between two points in the work of Ackermann, Blömer and Sohler [3] required a constant time. In our case, the distances are computed between two curves, where one has complexity at most m and another the complexity at most ℓ. Thus using Alt and Godau's algorithm (Theorem 2.32) for distance computations we require time $O(m\ell \log(m\ell))$ per distance computation, as an additional multiplicative factor to the running time of the algorithm of Theorem 4.21. Since $k, \ell, \epsilon, \gamma$ are constants, we can hide them in the O-notation, and obtain the following theorem, which is the main result of Section 4.5.

[13] The curves from P cannot be chosen to be the cluster centers (unless they have the complexity at most ℓ). However, the analysis of Theorem 4.20 guarantees the cluster centers to be from Δ_ℓ.

Theorem 4.22. *Let $0 < \varepsilon < 1$ and $k, \ell \in \mathbb{N}$ be constants. Given a set of curves $P = \{\tau_1, \ldots, \tau_n\} \subset \Delta_m$, there exists an algorithm that with constant probability returns a $(1+\varepsilon)$-approximation to $\mathrm{opt}^{(1)}_{k,\ell}(P)$ and a witness solution for input instance P, and that has running time $O(mn \log m)$.*

4.6 Hardness of clustering under the Fréchet distance

In this section we show that the (k, ℓ)-center and the (k, ℓ)-median clustering problems are **NP**-hard (if k is a part of the input). We reduce these problems to their classical counterparts (cf. Section 2.4) under ℓ_p-norms in $\mathbb{R}^d$.

The hardness of both (k, ℓ)-clustering problems for $\ell \geq 6$ follows from the following lemma, stated in the survey of Indyk and Matoušek [113]. The lemma is also valid for the case of the discrete Fréchet distance.

Lemma 4.23 (cf. [113])**.** *One can isometrically embed any bounded subset of a d-dimensional vector space equipped with the ℓ_∞-norm into Δ_{3d} equipped with the continuous Fréchet distance .*

This immediately implies **NP**-hardness for $\ell \geq 6$ knowing that the clustering problems we consider are **NP**-hard under the ℓ_∞ distance for $d \geq 2$. We want to extend the **NP**-hardness result to hold for $\ell \geq 2$. This is achieved by preserving $\ell = d$ in the metric embedding. The following lemma describes the needed embedding.

Lemma 4.24. *Any metric space $(\mathcal{X}, \ell_\infty)$, where $\mathcal{X} \subset \mathbb{R}^d$ is a bounded set, can be embedded isometrically into (Δ_d, d_F). Furthermore, if $\mathcal{X}$ is discrete, the embedding and its inverse can be computed in time linear in $|\mathcal{X}|$ and d.*

Proof. Given bounded set $\mathcal{X} \subset \mathbb{R}^d$. We denote the i-th coordinate of $x \in \mathcal{X}$ with x_i. Let $s' = \min_{x \in \mathcal{X}} \{x_i \colon i \in [d]\}$ and $s'' = \max_{x \in \mathcal{X}} \{x_i \colon i \in [d]\}$. Let $\vartheta = s'' - s'$. We define the embedding $f \colon \mathcal{X} \to \Delta_d$ as follows. To each coordinate x_i, $i \in [d]$, we assign the vertex $x'_i = x_i + (-1)^i \cdot 3\vartheta$. The curve $f(x) \in \Delta_d$ is obtained by a linear interpolation of the vertices x'_i in order of the coordinate index i. The vertices of curves $f(x)$ are alternating local minima and maxima, and for each two consecutive vertices x'_i, x'_{i+1} of $f(x)$, $i \in [d-1]$, it holds that $5\vartheta \leq |x'_{i+1} - x'_i| = |x_{i+1} - x_i + 6\vartheta| \leq 7\vartheta$.

It is clear that for any $x, y \in \mathcal{X}$ it is $d_F(f(x), f(y)) \leq \|x - y\|_\infty \leq \vartheta$, since we can map the vertices of $f(x)$ and $f(y)$ bijectively by mapping x'_i to y'_i, $i \in [d]$, and such mapping witnesses the Fréchet distance of at most $\|x - y\|_\infty$. Let us assume for the sake of contradiction, that $\delta = d_F(f(x), f(y)) < \|x - y\|_\infty$. Then, for each curve $f(x)$ it holds that it is its own δ-signature. Since $d_F(f(x), f(y)) = \delta$, then by Lemma 3.5 there has to

be a vertex of $f(y)$ in each range $R_i^{(x)} = [x'_i - \delta, x'_i + \delta]$, and these vertices have to appear on $f(y)$ in the order of i. Analogously, there has to be a vertex of $f(x)$ in each range $R_i^{(y)} = [y'_i - \delta, y'_i + \delta]$.

No two consecutive vertices y'_j, y'_{j+1} of $f(y)$, (respectively, x'_i, x'_{i+1} of $f(x)$) being at distance of at least 5ϑ, can lie in the same range $R_i^{(x)}$ (respectively $R_j^{(y)}$). Since both $f(x)$ and $f(y)$ have complexity d, for each vertex x'_i, $i \in [d]$, the vertex y'_i has to lie in $R_i^{(x)}$. But then for the index j, such that $|x_j - y_j| = \|x - y\|_\infty$, would hold that $\delta \geq \|x - y\|_\infty$, a contradiction.

If $\mathcal{X}$ is discrete it is clear from the construction, that both the embedding and its inverse can be computed in time linear in $|\mathcal{X}|$ and d. This closes the proof. □

The previous proof holds for the embedding into (Δ_d, d_{dF}) as well. We cannot use Lemma 3.5, but since the discrete Fréchet distance maps vertices to vertices of the two curves, by having that $\delta = d_{dF}(f(x), f(y)) < \|x-y\|_\infty$ there would need to exist a vertex y'_j in each range $R_i^{(x)}$, so that the pair (i, j) is in the traversal that witnesses $d_{dF}(f(x), f(y))$. The rest of the proof holds verbatim.

The **NP**-hardness reduction takes an instance of the k-center (respectively, k-median) problem under ℓ_∞ in $\mathbb{R}^d$ and embeds it into Δ_d (under continuous or discrete Fréchet distance) using Lemma 4.24. If we could solve the (k, d)-center (respectively, (k, d)-median) problem (for definitions see Subsection 4.1.1), then by Lemma 4.24, we could apply the inverse embedding function to the solution to obtain a solution for the original problem instance. The same holds for the approximate solution.

Note that the embedding given in Lemma 4.24 works for any point in the convex hull of $\mathcal{X}$, therefore also for the centers (respectively, medians) that form the solution.

The following theorems state our **NP**-hardness results.

Theorem 4.25. *The (k, ℓ)-center problem (where k is part of the input) under both continuous and discrete Fréchet distance is* **NP***-hard for $\ell \geq 2$. Furthermore, the problem is* **NP***-hard to approximate within a factor of* 2.

Theorem 4.26. *The (k, ℓ)-median problem (where k is part of the input) under both continuous and discrete Fréchet distance is* **NP***-hard for $\ell \geq 2$.*

4.7 Conclusion and open questions

The (k, ℓ)-clustering problems are still open in several cases, and they offer several possibilities for further extensions and a potential research. For the two problems we considered in

this chapter the hardness and many approximation algorithms we discussed in Section 4.1.3 are known. However, no $(1+\varepsilon)$-approximation algorithm for the (k,ℓ)-center problem for higher dimensions under the continuous Fréchet distance is known yet.

For the (k,ℓ)-median problem under the continuous Fréchet distance our result (for the one-dimensional ambient space) is the only known $(1+\varepsilon)$-approximation algorithm, for the problem without any restrictions. It is probable that for an extension into the multidimensional ambient space, and in particular for a $(1+\varepsilon)$-approximation, the algorithm of Ackermann, Blömer and Sohler [3] would be used as a tool, as we did in the one-dimensional case. This result was used (in an adapted form) by Nath and Taylor [146] for their $(1+\varepsilon)$-approximation algorithm in the discrete Fréchet distance case, as well as by Meintrup, Munteanu and Rohde [139] and by Buchin, Driemel and Rohde [46] in the continuous Fréchet distance case.

In this thesis we did not consider two related clustering problems with bounded complexity of the cluster center curves: (k,ℓ)-means, which extends the k-means problem in general metric spaces (cf. Equation (2.23)), and the (k,ℓ)-clustering under DTW distance, where in our definitions the Fréchet distance is replaced with DTW distance.

For the (k,ℓ)-means clustering, using notation of Subsection 4.1.1, we would aim to minimize the function

$$\mathrm{cost}_2(P,C) = \sum_{i=1}^{n} \left[\min_{j\in\{1,\dots,k\}} d_F(\tau_i,\varsigma_j)\right]^2,$$

analogously to the definitions of our problems in Subsection 4.1.1. A potential problem is that the dissimilarity measure defined as the squared metric distance is no longer a metric, since it does not satisfy the triangle inequality, but a weaker version of it. We can adapt our constant-factor approximation algorithm for the (k,ℓ)-clustering to the (k,ℓ)-means problem under both discrete and continuous Fréchet distance. Such an adaptation of Algorithm 2 produces a $(2\alpha^2+4\beta+4\alpha^2\beta)$-approximation algorithm, where $\alpha=2$ remains the simplification step factor as in the (k,ℓ)-median case. For the clustering step factor β we can use the 6.357-approximation algorithm by Ahmadian *et al.* [11], or the $9+\varepsilon$-approximation algorithm by Kanungo *et al.* [117] for the k-means clustering problem in general metric spaces. The running times of these algorithm is polynomial in k and n (in the case of Kanungo *et al.* it is $O\left(n^3\varepsilon^{-d}\right)$). Thus, such an algorithm would not have a near-linear time in terms of the input, as opposed to the algorithm for our two aforementioned clustering problems, and a much weaker approximation factor.

For the (k,ℓ)-clustering under the dynamic time warping distance, a drawback is that the DTW distance is not a metric. But it is possible that, as we did for the (k,ℓ)-

median problem, the dissimilarity measure d_{DTW} can be incorporated into the analysis of Ackermann, Blömer and Sohler [3], provided that the complexity of the candidate center curves is bounded with ℓ, and that the sampling property (Theorem 4.14) is satisfied. The only related result is the work of Brill *et al.* [30], that gave an exact computation algorithm for the 1-median problem under DTW, but this result holds only for one-dimensional ambient space and has a running time exponential in the complexity of the input curves.

5 Embedding of the Fréchet distance

5.1 Introduction

In Chapter 4 we have seen that the clustering problems under the Fréchet distance can be efficiently approximated for one-dimensional curves, but for higher dimensions these problems are harder and there are even more open questions. Buchin *et al.* [43] observed that the problem of computing the continuous Fréchet distance has a special structure in the one-dimensional space, and there is no known lower bound for this problem. Bringmann and Künnemann [34] used projections of the curves to the one-dimensional space to speed up their approximation algorithm for the Fréchet distance computation. It is tempting to assume that if we would restrict the curves to the one-dimensional ambient space, then the continuous Fréchet distance computation problem would be significantly simplified. However, this is not necessarily true, as in the general case there is no known algorithm to compute the continuous Fréchet distance that performs better on one-dimensional than on multi-dimensional curves.

It is conventional practice to separate the coordinates of the curves' vertices to simplify computational tasks. It seems that the inherent character of a curve is often largely preserved if restricted to one of coordinates of the ambient space. This is equivalent to embedding the curves to the space of one-dimensional curves by projecting them to a line. This is one motivation for the study presented in this chapter.

Another motivational factor comes from the work of Driemel and Silvestri [71] on probabilistic data structures, in particular locally sensitive hashing (LSH) functions for the discrete Fréchet distance. There are no known LSH functions for the continuous case. It is conceivable that the concept of signatures we discussed in Chapter 3 together with projections of the curves to random lines could be used for defining an LSH function. Therefore, in this chapter we study the distortion of the probabilistic embedding of the Fréchet distance between two polygonal curves that results from projecting them to a randomly chosen line.

5.1.1 Problem definition

Consider two polygonal curves $\sigma = v_1, v_2, \ldots, v_m$ and $\tau = w_1, w_2, \ldots, w_m$ with m vertices, each in $\mathbb{R}^d$, given by their sequences of vertices. Consider sampling a unit vector $\mathbf{u}$ in respective $\mathbb{R}^d$ by choosing uniformly at random a point on the unit $(d-1)$-sphere (the surface of the d-ball in $\mathbb{R}^d$) centered at the origin. We denote with L the line through the origin that supports the vector $\mathbf{u}$. Let $\sigma' = v'_1, v'_2, \ldots, v'_m$ and $\tau' = w'_1, w'_2, \ldots, w'_m$ be the projections of σ and τ to L. If we denote $\mathbf{v}_i$, $\mathbf{w}_j$, $\mathbf{v}'_i$, and $\mathbf{w}'_j$ the position vectors in $\mathbb{R}^d$ associated with respective points v_i, w_j, v'_i, and w'_j, for all $1 \leq i \leq m$ and $1 \leq j \leq m$, then these projections are defined by $\mathbf{v}'_i = \langle \mathbf{v}_i, \mathbf{u} \rangle \, \mathbf{u}$ and $\mathbf{w}'_j = \langle \mathbf{w}_j, \mathbf{u} \rangle \, \mathbf{u}$.

Note that throughout this chapter, for $x, y \in \mathbb{R}^d$ we denote the Euclidean distance between the points x and y with $\|x - y\|$. If $\mathbf{x}$ and $\mathbf{y}$ are their respective associated vectors in $\mathbb{R}^d$, then $\|x - y\|$ equals the Euclidean norm of the vector $\|\mathbf{x} - \mathbf{y}\|$. This distinction is only used in Lemma 5.1, Lemma 5.3, and Lemma 5.5, and for the rest of the chapter we can overload the norm notation.

Since the (continuous and discrete) Fréchet distance, as well as the dynamic time warping distance, always decrease when the curves are projected to a line (cf. Lemma 5.3), we ask which extent this decrease can have, for the general case curves and in particular for the realistic class of input curves – c-packed curves (cf. Definition 2.30). We are interested both in upper and lower bounds.

5.1.2 Results in this chapter

We start our problem analysis by stating several basic results on projections of the points and the curves from the d-dimensional to a one-dimensional Euclidean space. This is presented in Section 5.2.

When computing the upper bound, we assume that $d \in \{2, 3, 4, 5, 6, 7\}$. In Section 5.3 we show that if the curves σ and τ are c-packed for constant c, then, with constant probability, the discrete Fréchet distance between the curves σ and τ, denoted by $d_{dF}(\sigma, \tau)$, degrades by at most a linear factor in m. This result is presented by Theorem 5.15. To obtain this theorem, we explore the properties and the inner structure of the free-space matrix of two curves.

In Section 5.4 we consider the lower bounds on our problem. For the c-packed curves, the upper bound on the ratio of the two distances is matched by a lower bound that is also linear in m. This lower bound result is presented by Theorem 5.16. The construction of the lower bound uses c-packed curves with $c < 3$. Theorem 5.16 holds for the continuous Fréchet distance and for the dynamic time warping distance as well.

We also show that there exist polygonal curves σ and τ that are not c-packed for sublinear c and their (continuous or discrete) Fréchet distance degrades by a linear factor for any projection line (i.e. with probability 1). Theorem 5.17 presents this result.

5.1.3 Related work

Embedding of the metric spaces into low-dimensional geometric spaces is a fundamental problem, whose research roots back to the middle of the 20th century. It has multiple algorithmic applications and for an overview we refer to the survey of Indyk [111]. We discuss what is known for two variations of the metric embedding problem that are most studied, emphasizing the results for embedding into a line.

The first problem is to find the smallest distortion for any metric from the given class. This problem is called a *combinatorial problem* by Sidiropoulos *et al.* [160]. Matoušek [135] showed that any metric on a point set of size s can be embedded into d-dimensional Euclidean space with multiplicative distortion $O\left(\min\{s^{2/d}\log^{3/2} s, s\}\right)$, but not better than $\Omega\left(s^{1/\lfloor(d+1)/2\rfloor}\right)$. This implies that for $d = 1$ the distortion is linear in the worst case.

The second problem, called an *algorithmic problem* by Sidiropoulos *et al.* [160], is to find the smallest approximation factor to a minimal distortion for a given metric over a point set $\mathcal{X}$ of size s. Matoušek [134] showed that any shortest path metric on a graph $G = (V, E)$ with s vertices can be embedded into a line with distortion at most $2s - 1$ in time $O(|V| + |E|)$.

We call the maximum/minimum ratio of the distances of the input point set $\mathcal{X}$ the **spread** Ψ. Bădoiu *et al.* [19] gave an $O\left(\Psi^{3/4}\mathfrak{c}^{11/4}\right)$-approximation to the embedding to a line, where $\mathfrak{c}$ is the distortion of embedding of the input set onto the line. They also showed that it is hard to approximate this problem up to a factor $\Omega\left(s^{1/12}\right)$, even for a weighted tree metrics with polynomial spread. Assuming a constant distortion $\mathfrak{c}$ and a polynomial spread Ψ, Nayyeri and Raichel [147] gave an $O(1)$-approximation algorithm to the minimal distortion of the embedding to a line, in time polynomial in s and Ψ.

Håstad, Ivansson and Lagergren [99] studied the matrix-to-line problem, i.e. given s points and given their distances in a symmetric matrix, the aim is to find an embedding of the points into a line, such that the distances of the embedded points agree as much as possible to the original distances. Their distortion is defined as a maximum *difference* between the distance of embedded points $|f(x) - f(y)|$ and the original distance $\mathbf{d}(x, y)$, for two input points x and y. They gave a 2-approximation algorithm for that problem, but also showed that it is **NP**-hard to approximate better than a factor of $7/5$.

Sidiropoulos *et al.* [160] considered the problem of a noncontracting embedding f of a graph G with s vertices (that induces a shortest path metric $\mathbf{d}$) into a line, such that

the distortion $\mathfrak{c}$ is minimized. Their distortion is defined as a maximum *ratio* between the distance of embedded points $|f(x) - f(y)|$ and the original distance $\mathbf{d}(x, y)$, for $x, y \in G$. Their algorithm gave an $O(\mathfrak{c})$-approximation for metrics whose distortion is at most $\mathfrak{c}$, and an $O(\sqrt{s})$-approximation for general metrics. They gave an exact algorithm that requires $O(n^{\mathfrak{c}_{\text{opt}}})$ time, where $\mathfrak{c}_{\text{opt}}$ is the optimal distortion. This result that can be paired with their proof that an α-approximation of $\mathfrak{c}_{\text{opt}}$ is **NP**-hard for certain $\alpha > 1$, where the factor α comes from the **NP**-hardness result on the travelling salesman problem.

Fellows *et al.* [82] showed that given an unweighted graph G with s vertices, and a positive integer $\mathfrak{c}$, it is possible in time $O\left(s \cdot \mathfrak{c}^4(2\mathfrak{c}+1)^{2\mathfrak{c}}\right) = O(s) \cdot 2^{O(\mathfrak{c} \log \mathfrak{c})}$ either to embed the shortest path metric defined by the graph G into the real line with distortion at most $\mathfrak{c}$, or to conclude correctly that no such embedding exists. Thus the running time of their algorithm is linear for every fixed $\mathfrak{c}$, and the problem is fixed parameter tractable (FPT), parameterized by the distortion. For this problem it was shown by Lokshtanov, Marx and Saurabh [133] that the dependency on $\mathfrak{c}$ cannot be reduced to $2^{o(\mathfrak{c} \log \mathfrak{c})}$ unless Exponential Time Hypothesis (ETH) fails.[14]

It is not much known on embeddings under the Fréchet distance. The result that is closest comparable to that we develop and present in this chapter was given by Bačkurs and Sidiropoulos [18]. They gave an embedding of the Hausdorff distance into constant-dimensional ℓ_∞-space with constant distortion. More precisely, for any $s, d \geq 1$, they obtained an embedding for the Hausdorff distance over point sets of size s in d-dimensional space, into $\ell_\infty^{s^{O(s+d)}}$ with distortion $s^{O(s+d)}$. No such metric embeddings are known for the discrete or continuous Fréchet distance.

By extending the random projection $(1+\varepsilon)$-embedding of Johnson and Lindenstrauss [116] to the case of n curves under the continuous Fréchet distance in the d-dimensional space and under assumptions on the length of the edges of the curves, Meintrup, Munteanu and Rohde [139] obtained an embedding into $O\left(\varepsilon^{-2} \log n\right)$-dimensional space, which has an additive (and not multiplicative) error component. It is not clear if such an approach can yield a multiplicative $(1+\varepsilon)$-approximation of the Fréchet distance, with or without additional assumptions.

Since the doubling dimension of spaces equipped with the Fréchet distance is unbounded, even for the case when the metric space is restricted to curves of constant complexity, as shown in Section 4.2, a result of Bartal, Gottlieb and Neiman [24] for spaces with finite doubling dimension implies that a metric embedding of the Fréchet distance into an ℓ_p-space would have at least super-constant distortion. However, it is not known how to find such an embedding.

[14] Cf. Hypothesis 2.34 for the distinction between ETH and SETH.

The complexity of classic data structuring problems for the Fréchet distance is an active research topic. Since we intended to develop an embedding technique, that would be useful for the nearest-neighbor searching and range searching problems, we review next what is known about them. An α-approximate nearest-neighbor data structure in a metric space $(\mathcal{X}, \mathbf{d})$ returns, for a given data point set $S \subseteq \mathcal{X}$ and a given query point q, a data point $p \in S$, such that the distance $\mathbf{d}(p, q)$ is at most $\alpha \cdot \mathbf{d}(p^*, q)$, where $p^* \in S$ is the true nearest neighbor to q. Indyk [112] gave a deterministic and approximate near-neighbor data structure for the discrete Fréchet distance, using an embedding of the metric space with the discrete Fréchet distance into an inner product space. Indyk's data structure for data set S, containing n curves which have at most m vertices, achieves approximation factor $c \in O(\log m + \log\log n)$ and has query time $O(\text{poly}(m) \cdot \log n)$. This data structure requires very large space (exponential in $\sqrt{m}$), as it precomputes all queries with curves with $\sqrt{m}$ vertices.

For short curves (with $m \in O(\log n)$) Driemel and Silvestri [71] described an approximate near-neighbor structure for the discrete Fréchet distance, based on locality-sensitive hashing (LSH) with approximation factor $O(m)$, query time $O(m \log n)$, and using space $O(n \log n + mn)$. An experimental evaluation of the data structure of Driemel and Silvestri with improvements was presented by Ceccarello, Driemel and Silvestri [51]. LSH is a technique that uses families of hash functions with the property that near points are more likely to be hashed to the same index than far points. Driemel and Silvestri were the first to define locality-sensitive hash functions for the discrete Fréchet distance. Emiris and Psarros [76] improved their result and also showed how to obtain $(1+\varepsilon)$-approximation with query time $\tilde{O}(d \cdot 2^{2m} \cdot \log n)$ using space $\tilde{O}(n) \cdot (2 + d/\log m)^{O(m \cdot d \cdot \log(1/\varepsilon))}$. No such hash functions are known for the continuous case.

Only recently two $(1+\varepsilon)$-approximate near neighbor data structures for the discrete Fréchet distance in d-dimensional ambient space were given, both following the approach of preprocessing the answers to all relevant queries on a discretization of the space. Those are results of Filtser, Filtser and Katz [84] and Driemel, Psarros and Schmidt [70]. Both papers considered the asymmetric setting where the query curves have much smaller complexity $\ell \ll m$ than the input curves. Driemel, Psarros and Schmidt presented a construction of a randomized data structure that uses space $O\left(n \cdot \left(d^{3/2}\ell\varepsilon^{-1}\right)^{d\ell}\right)$ and needs query time $O(d\ell)$. They also gave a derandomized algorithm, which causes an increase of the space used to $O\left(n \cdot d^{3/2}\ell\varepsilon^{-1} \cdot \left(d^{3/2}\ell\varepsilon^{-1}\right)^{d\ell}\right)$ and the query time to $O\left(d^{5/2}\ell^2\varepsilon^{-1} \cdot \left(\log n + d\ell \log\left(d\ell\varepsilon^{-1}\right)\right)\right)$. The most recent version of the work of Filtser, Filtser and Katz presents the data structure that needs space $O\left(n \cdot \left(\varepsilon^{-1}\right)^{d\ell}\right)$ and query time $O(d\ell)$ in randomized version and $O\left(d\ell \log\left(nd\ell\varepsilon^{-1}\right)\right)$ in derandomized version. Their

result extends to the dynamic time warping distance. These results suggest, that the preprocessing approach is more efficient than the approaches that use LSH or randomized projections.

Until recently, there were no results known for the approximate near-neighbor problem under the continuous Fréchet distance, besides of using the discrete Fréchet distance to approximate the continuous Fréchet distance. The recent result of Driemel and Psarros [69] gives a $(2+\varepsilon)$-approximation solution to the problem in one-dimensional ambient space $\mathbb{R}$, for the query curve of complexity ℓ. Their data structure uses space $O\left(n \cdot (1/\varepsilon)^{\ell} + mn\right)$ and has query time $O(\ell)$, after the preprocessing that needs $O\left(n \cdot (1/\varepsilon)^{\ell} + mn\ell^3\right)$ time.

The range searching problem under the Fréchet distance receives a set S of n curves as an input, each of complexity m in the ambient space $\mathbb{R}^d$. The goal is to report all curves from the input, such that their (continuous or discrete) Fréchet distance to the query curve of complexity ℓ is at most some threshold value $\delta \geq 0$. A related problem of range counting is to answer how many distinct subcurves are within a given threshold to a query curve. For both problems the challenge is to build a data structure, such that answering the queries is efficient.

De Berg *et al.* [61] studied range counting data structures for spherical range search queries under the continuous Fréchet distance, assuming that the query curves are line segments. They built a data structure that stores compressed subcurves of a single polygonal curve, and utilizing a partition tree. Their data structure uses space $O(s \cdot \text{polylog}(n))$ and has query time $O((n/\sqrt{s}) \cdot \text{polylog}(n))$ to obtain a constant-approximation factor solution, where $n \leq s \leq n^2$ is a parameter of the data structure which is fixed at the preprocessing time.

Afshani and Driemel [4] showed how to leverage semi-algebraic range searching for the range searching problem under the Fréchet distance. Their data structure supports polygonal curves of low complexity and answers queries exactly. In particular, for the discrete Fréchet distance they described a data structure which uses space in $O\left(n \cdot (\log\log n)^{m-1}\right)$ and achieves query time in $O\left(n^{1-1/d} \cdot \log^{O(m)} n \cdot \ell^{O(d)}\right)$, where it is assumed that the complexity of the query curves ℓ is upper-bounded by a polynomial of $\log n$. For the continuous Fréchet distance they described a data structure for polygonal curves in the plane which uses space in $O\left(n \cdot (\log\log n)^{O(m^2)}\right)$ and achieves query time in $O\left(\sqrt{n} \cdot \log^{O(m^2)} n\right)$. For the case where the curves lie in dimension higher than 2 and the distance measure is the continuous Fréchet distance, no data structures for range searching or range counting are known.

5.2 Preliminaries

Given curves $\sigma = v_1, v_2, \ldots, v_m$ and $\tau = w_1, w_2, \ldots, w_m$, we denote $\delta_{i,j} = \|v_i - w_j\|$ and $\delta'_{i,j} = \|v'_i - w'_j\|$, for all $1 \leq i \leq m$ and $1 \leq j \leq m$, i.e. $\delta_{i,j}$ and $\delta'_{i,j}$ are the pairwise distances of the vertices for the input curves σ and τ and for their respective projections σ' and τ'.

Furthermore, we define a directed, vertex-weighted graph $G = (V, E)$ on the node set $V = \{(i, j) : 1 \leq i, j \leq m\}$. A node (i, j) corresponds to a pair of vertices v_i of σ and w_j of τ and we assign it the weight $\delta_{i,j}$. The set of edges is defined as $E = \{((i, j), (i', j')) : i' \in \{i, i+1\}, j' = \{j, j+1\}, 1 \leq i, i', j, j' \leq m\}$. The set of paths in the graph G between $(1, 1)$ and (m, m) corresponds to the set of traversals $\mathcal{T}$ of σ and τ. A path in G which does not start in $(1, 1)$ or end in (m, m) is called a **partial traversal** of σ and τ.

It is useful to picture the nodes of the graph G as a matrix, where rows correspond to the vertices of σ and columns correspond to the vertices of τ. For any fixed value $\Theta > 0$, we define the **free-space matrix** $F_\Theta = (\phi_{i,j})_{1 \leq i,j \leq t}$ with

$$\phi_{i,j} = \begin{cases} 1 & \text{if } \|v_i - w_j\| < \Theta \\ 0 & \text{if } \|v_i - w_j\| \geq \Theta. \end{cases}$$

Note that the conventional definition of the free-space matrix for parameter Θ, analogous to the definition of the Θ-free-space of Equation 2.27 in Subsection 2.5.3 is slightly different. There was an 1-entry in the free-space matrix if and only if $\|v_i - w_j\| \leq \Theta$. We adapt the definition in this chapter since it better suits our needs.

Overlaying the graph with the free-space matrix for $\Theta > d_{dF}(\sigma, \tau)$, we can observe that there exists a path in the graph from $(1, 1)$ to (m, m) that visits only the matrix entries with value 1. Moreover, the existence of such a path in the free-space matrix for some value of Θ implies that $\Theta > d_{dF}(\sigma, \tau)$.

We prove the following basic fact about random projections to a line, stated for $d \in \{2, 3, 4, 5, 6, 7\}$ by Lemma 5.1. After the proof of the lemma we discuss briefly what happens in higher dimensions.

Lemma 5.1. *If two points p and q are projected to the straight line L, which supports the unit vector chosen uniformly at random on the unit sphere in $\mathbb{R}^d$, $d \in \{2, 3, 4, 5, 6, 7\}$, the probability that the distance of their projections will be reduced from the original distance by a factor greater than φ is at most $e \cdot \varphi$, where e is a constant. The constant e equals 1 for $d \in \{2, 3\}$, $1 + 2/\pi$ for $d \in \{4, 5\}$, and $15/8$ for $d \in \{6, 7\}$.*

Proof. Let p and q be two vertices in $\mathbb{R}^d$. Let $\mathbf{u}$ be the unit vector chosen uniformly at random on the unit sphere in $\mathbb{R}^d$, and let L be the straight line that supports the vector $\mathbf{u}$. Then let p' and q' be the projections of p and q respectively to the projection line L. Let $\mathbf{p}$, $\mathbf{q}$, $\mathbf{p}'$, and $\mathbf{q}'$ be the vectors associated with vertices p, q, p', and q' respectively. Let α be the angle between $\mathbf{u}$ and the vector $\mathbf{q} - \mathbf{p}$ (cf. Figure 5.1). Then it holds by the definion of the inner product that

$$\|\mathbf{q}' - \mathbf{p}'\| = \| \langle \mathbf{q} - \mathbf{p}, \mathbf{u} \rangle \cdot \mathbf{u}\| = \|\mathbf{q} - \mathbf{p}\| \cdot \|\mathbf{u}\| \cdot |\cos\alpha| \cdot \|\mathbf{u}\|. \tag{5.1}$$

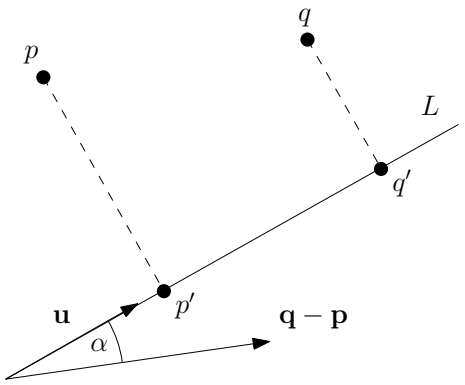


Figure 5.1: The projection of the pair of the vertices to the straight line

Since the projection line L supports the vector $\mathbf{u}$, which is chosen uniformly at random on the unit $(d-1)$-sphere in $\mathbb{R}^d$ centered at origin, the angle α is distributed by the probability distribution function $h_d(\alpha)$, defined as the ratio of the surface of a $(d-2)$-sphere ($(d-1)$-dimensional ball) of radius $\sin\alpha$, and the surface of a unit $(d-1)$-sphere (d-dimensional ball) (cf. Lemma 2.11). Using Equation (2.6), this can be expressed as:

$$h_d(\alpha) = \frac{1}{\sqrt{\pi}} \cdot \frac{\Gamma\left(\frac{d}{2}\right)}{\Gamma\left(\frac{d-1}{2}\right)} \cdot (\sin\alpha)^{d-2} \tag{5.2}$$

over the interval $\alpha \in [0, \pi]$.

Since $\sin(\pi/2 \pm \alpha') = \cos\alpha'$, it is $h_d(\pi/2 + \alpha') = h_d(\pi/2 - \alpha')$, for $\alpha' \in [0, \pi/2]$, i.e. all functions $h_d(\alpha)$ are symmetric around $\pi/2$. It is $|\cos\alpha| \geq \varphi$ for $\alpha \in [0, \arccos\varphi] \cup [\pi - \arccos\varphi, \pi]$, for any $\varphi \in [0, 1]$. Then from Equations (5.1) and (5.2) we have that for all $d \geq 2$ and $\varphi \in [0, 1]$ it is

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] = 1 - 2 \cdot \int_0^{\arccos\varphi} h_d(\alpha) d\alpha. \tag{5.3}$$

Case $d = 2$: The distribution of α in Equation (5.2) is uniform with $h_2(\alpha) = 1/\pi$. Thus Equation (5.3) implies

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] = 1 - \frac{2\arccos\varphi}{\pi}. \tag{5.4}$$

Using Taylor series of $\arccos\varphi$ we get for $0 \leq \varphi \leq 1$:

$$\begin{aligned}\arccos\varphi &= \frac{\pi}{2} - \sum_{k=0}^{\infty} \frac{(2k)! \cdot \varphi^{2k+1}}{2^{2k} \cdot (2k+1) \cdot (k!)^2} = \frac{\pi}{2} - \varphi - \sum_{k=1}^{\infty} \frac{(2k)! \cdot \varphi^{2k+1}}{2^{2k} \cdot (2k+1) \cdot (k!)^2} \\ &\geq \frac{\pi}{2} - \varphi - \varphi^3 \cdot \sum_{k=1}^{\infty} \frac{(2k)!}{2^{2k} \cdot (2k+1) \cdot (k!)^2} = \frac{\pi}{2} - \varphi - \varphi^3 \cdot \left(\frac{\pi}{2} - 1\right),\end{aligned}$$

since $\varphi \geq \varphi^3 \geq \varphi^{2k+1}$ for all $k \geq 1$. Therefore

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] = 1 - \frac{2\arccos\varphi}{\pi} \leq \frac{2}{\pi} \cdot \varphi + \left(1 - \frac{2}{\pi}\right) \cdot \varphi^3 \leq \varphi. \tag{5.5}$$

Case $d = 3$: The distribution of α is $h_3(\alpha) = (\sin\alpha)/2$, for $\alpha \in [0, \pi]$. Thus Equation (5.3) implies that

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] = 1 - 2 \cdot \int_0^{\arccos\varphi} \frac{\sin\alpha}{2} d\alpha = 1 - (1 - \varphi) = \varphi. \tag{5.6}$$

Case $d = 4$: The distribution of α is $h_4(\alpha) = \left(2\sin^2\alpha\right)/\pi$ for $\alpha \in [0, \pi]$. Thus

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] = 1 - 2 \cdot \int_0^{\arccos\varphi} \frac{2}{\pi}\sin^2\alpha d\alpha \overset{(A.1)}{=} 1 - \frac{2}{\pi}\left[\arccos\varphi - \varphi \cdot \sqrt{1 - \varphi^2}\right].$$

This expression implies, using the last two inequalities of (5.5), that

$$\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] \leq \varphi + \frac{2}{\pi} \cdot \varphi \cdot \sqrt{1 - \varphi^2} \leq \left(1 + \frac{2}{\pi}\right) \cdot \varphi. \tag{5.7}$$

Case $d = 5$: The distribution of α is $h_5(\alpha) = \left(3\sin^3\alpha\right)/4$ for $\alpha \in [0, \pi]$. We have that

$$\begin{aligned}\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] &= 1 - 2 \cdot \int_0^{\arccos\varphi} \frac{3}{4}\sin^3\alpha d\alpha \overset{(A.3)}{=} 1 - \frac{3}{2}\left(\frac{1}{3}\varphi^3 - \varphi + \frac{2}{3}\right) \\ &\overset{(A.12)}{\leq} \left(1 + \frac{2}{\pi}\right) \cdot \varphi.\end{aligned} \tag{5.8}$$

Case $d = 6$: The distribution of α is $h_6(\alpha) = \left(8 \sin^4 \alpha\right)/(3\pi)$ for $\alpha \in [0, \pi]$. Then (5.3) implies

$$\begin{aligned}\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] &= 1 - 2 \cdot \int_0^{\arccos \varphi} \frac{8}{3\pi} \sin^4 \alpha d\alpha \\ &\overset{(A.6)}{=} 1 - \frac{2}{3\pi}\left(3 \arccos \varphi + \varphi \cdot \sin(\arccos \varphi) \cdot \left(2\varphi^2 - 5\right)\right) \\ &\overset{(A.13)}{\leq} \frac{16}{3\pi}\varphi. \end{aligned} \tag{5.9}$$

Case $d = 7$: The distribution of α is $h_7(\alpha) = \left(15 \sin^5 \alpha\right)/16$ for $\alpha \in [0, \pi]$. Then Equation (5.3) implies

$$\begin{aligned}\mathbf{Pr}\left[\frac{\|q' - p'\|}{\|q - p\|} < \varphi\right] &= 1 - 2 \cdot \int_0^{\arccos \varphi} \frac{15}{16} \sin^5 \alpha d\alpha \\ &\overset{(A.9)}{=} 1 - \frac{1}{8}\left(-3\varphi^5 + 10\varphi^3 - 15\varphi + 8\right) \overset{(A.14)}{\leq} \frac{15}{8}\varphi. \end{aligned} \tag{5.10}$$

For the extensive analysis confer to Appendix A. This closes the proof of the lemma. □

For a general problem in much higher dimension d, the probability stated by Lemma 5.1 cannot be bounded by a linear function in φ. Figure 5.2 shows the probability distribution function $h_d(\alpha)$ of Equation (5.2) for chosen values of d.

Notice that with increase of the dimension d the probability concentrates around $\pi/2$. This is actually a well-known fact about the unit d-ball in high-dimensional spaces (cf. [28]): most of its volume is concentrated near its "equator". If we set the coordinate system in such a manner, that the first coordinate x_1 is in the direction of the "north" (say, the vector $\mathbf{q} - \mathbf{p}$), then the following lemma holds.

Lemma 5.2 ([28] Theorem 2.7)**.** *For $a \geq 1$ and $d \geq 3$, at least a $\left(1 - (2/a) \cdot e^{-a^2/2}\right)$ fraction of the volume of the d-dimensional unit ball has $|x_1| \leq a/\sqrt{d-1}$.*

For the sake of completeness we prove the following lemma, that also holds if the discrete Fréchet distance is replaced by the continuous Fréchet distance, or by the dynamic time warping distance.

Lemma 5.3. *Given two curves $\sigma = v_1, \ldots, v_m$ and $\tau = w_1, \ldots, w_m$ in $\mathbb{R}^d$, and let $\sigma' = v'_1, \ldots, v'_m$ and $\tau' = w'_1, \ldots, w'_m$ respectively, be their projections to the straight line L which supports the vector $\mathbf{u}$ chosen uniformly at random on the unit sphere in $\mathbb{R}^d$. It holds that $d_{dF}(\sigma, \tau) \geq d_{dF}(\sigma', \tau')$.*

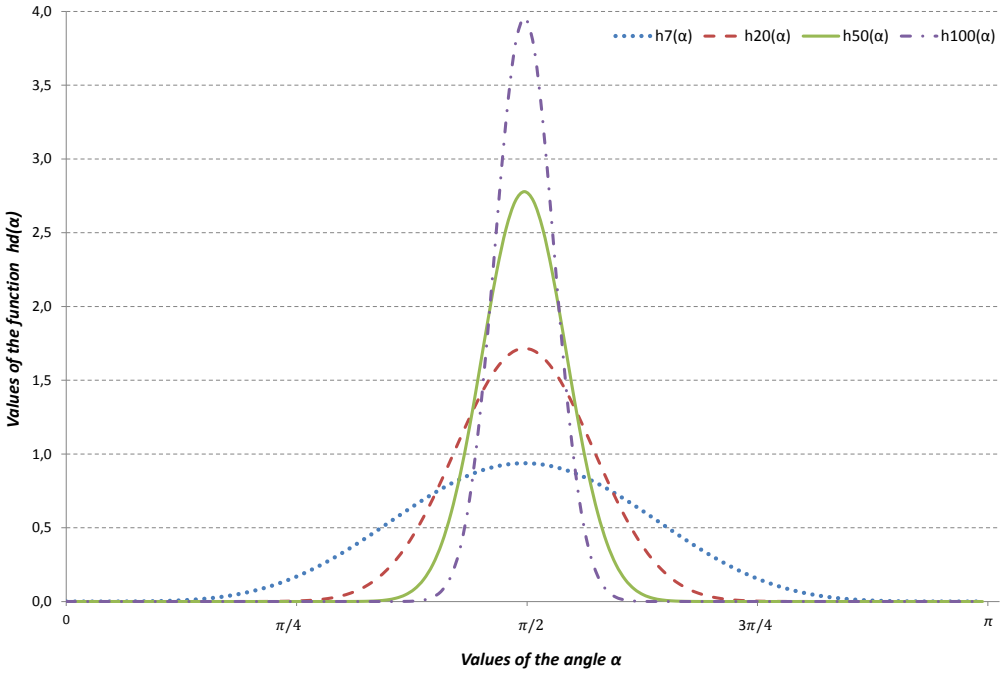


Figure 5.2: The distribution of the functions $h_d(\alpha)$, for $d \in \{7, 20, 50, 100\}$

Proof. Assume $d_{dF}(\sigma, \tau) < d_{dF}(\sigma', \tau')$ for some projection line L, and let T and T' be the traversals of σ and τ, and σ' and τ' that realize $d_{dF}(\sigma, \tau)$ and $d_{dF}(\sigma', \tau')$, respectively. It is $T, T' \in \mathcal{T}$, where $\mathcal{T}$ is the set of all traversals of σ and τ (and also of σ' and τ'). Then by assumption it holds that

$$\max_{(i,j)\in T} \|v_i - w_j\| = d_{dF}(\sigma, \tau) < d_{dF}\big(\sigma', \tau'\big) = \max_{(i,j)\in T'} \|v'_i - w'_j\|. \tag{5.11}$$

Let $\alpha_{i,j}$ be the angle between the vectors $\mathbf{w}_j - \mathbf{v}_i$ and $\mathbf{w}'_j - \mathbf{v}'_i$ (the latter being parallel to $\mathbf{u}$), for all pairs $(i, j) \in T$. Since any traversal of σ' and τ' is a traversal of σ and τ, using Equation (5.1) yields

$$\max_{(i,j)\in T'} \|v'_i - w'_j\| \leq \max_{(i,j)\in T} \|v'_i - w'_j\| = \max_{(i,j)\in T} \|v_i - w_j\| \cdot |\cos \alpha_{i,j}| \leq \max_{(i,j)\in T} \|v_i - w_j\|, \tag{5.12}$$

a contradiction. □

Lemma 5.3 holds for the continuous Fréchet distance as well. To adapt the proof we observe the curves as functions $\sigma, \tau, \sigma', \tau' : [0,1] \to \mathbb{R}^d$. Let f and f' be the matchings of σ and τ, and σ' and τ', respectively. Equation (5.11) in the proof of Lemma 5.3 becomes $d_F(\sigma,\tau) = \max_{t\in[0,1]} \|\sigma(t) - \tau(f(t))\| < \max_{t\in[0,1]} \|\sigma'(t) - \tau'(f'(t))\| = d_F(\sigma',\tau')$. A reparametrization between σ' and τ' is a reparametrization between σ and τ, thus Equation (5.12) becomes $\max_{t\in[0,1]} \|\sigma'(t) - \tau'(f'(t))\| \leq \max_{t\in[0,1]} \|\sigma'(t) - \tau'(f(t))\| \leq \max_{t\in[0,1]} \|\sigma(t) - \tau(f(t))\|$, a contradiction.

Furthermore, Lemma 5.3 holds for the dynamic time warping distance as well. To see this, we can repeat the proof for the discrete Fréchet distance, while replacing the *maximum* over the pairs of a traversal by the *sum* over the same pairs in both Equation (5.11) and Equation (5.12). The rest of the proof holds verbatim.

5.3 Upper bound

5.3.1 Guarding sets

The discrete Fréchet distance between curves σ and τ is realized by a pair (v_i, w_j) of vertices $v_i \in \sigma$ and $w_j \in \tau$, being at the distance $\|v_i - w_j\| = \delta$. We would like to apply Lemma 5.1 to this pair of vertices to show that the distance is preserved up to some constant factor. However, it is possible that the pairwise distances in the projection are such that a cheaper traversal is possible that avoids the pair (v_i, w_j) altogether. Therefore, we will apply the lemma to a subset of pairs of vertices of σ and τ whose distance is large (e.g. larger than $\Theta = \delta/\theta$ for some small value of $\theta \geq 1$) and such that the chosen set forms a hitting set for the set of traversals $\mathcal{T}$. To this end we introduce the notion of the *guarding set* by the following definition.[15]

Definition 5.4 (**Guarding set**). *For any two polygonal curves $\sigma = v_1, \ldots, v_m$ and $\tau = w_1, \ldots, w_m$, and a given parameter $\theta \geq 1$, a θ-guarding set $B \subseteq V$ for σ and τ is a subset of the set of vertices of G that satisfies the following conditions:*

(i) (distance property) *for all $(i,j) \in B$, it holds that $\delta_{i,j} \geq d_{dF}(\sigma,\tau)/\theta$, and*

(ii) (guarding property) *for any traversal T of σ and τ, it is $T \cap B \neq \emptyset$.*

The set B "guards" every traversal of σ and τ in the sense that any path in G from $(1,1)$ to (m,m) has a non-empty intersection with B. In other words, B is a hitting set for the set of traversals $\mathcal{T}$.

For a guarding set B we define the subset of vertices $S_B \subseteq V$ that can be reached by a path in G starting from $(1,1)$ without visiting a vertex of B. We call the set S_B the **reachable**

[15] The guarding sets for two curves σ and τ exist independently of $d_{dF}(\sigma,\tau)$. For the construction confer to Lemma 5.6 and Algorithm 5.

area defined by B. We also define the subset of vertices $H_B = V \setminus (B \cup S_B)$. A guarding set B thus defines a vertex partition of the graph G into three subsets $V = S_B \cup B \cup H_B$.

We show the following simple lemma for $d \in \{2, 3, 4, 5, 6, 7\}$ using Lemma 5.1.

Lemma 5.5. *Given parameter $\theta \geq 1$, if B is a θ-guarding set for the given curves $\sigma = v_1, \ldots, v_m$ and $\tau = w_1, \ldots, w_m$ from $\mathbb{R}^d$, $d \in \{2, 3, 4, 5, 6, 7\}$, and if σ' and τ' are their projections to the straight line L, whose support unit vector $\mathbf{u}$ is chosen uniformly at random on the unit sphere in $\mathbb{R}^d$, then for any $\beta > 1$ it holds that*

$$\frac{d_{dF}(\sigma', \tau')}{d_{dF}(\sigma, \tau)} \geq \frac{1}{e \cdot \beta \cdot \theta \cdot |B|}$$

with positive constant probability at least $1 - 1/\beta$. In the upper inequality, e is a constant and it equals 1 for $d \in \{2, 3\}$, $1 + 2/\pi$ for $d \in \{4, 5\}$, and $15/8$ for $d \in \{6, 7\}$.

Proof. Let $\mathbf{u}$ be the unit vector which is chosen uniformly at random on the unit sphere in $\mathbb{R}^d$ with $d \in \{2, 3, 4, 5, 6, 7\}$, and let $\mathbf{u}$ be supported by the projection line L. Let $\alpha_{i,j}$ be the angle between $\mathbf{u}$ and the vector $\mathbf{w}_j - \mathbf{v}_i$, for $i, j \in \{1, \ldots, m\}$. If we consider the distances of the pairs of the points $(v_i, w_j) \in \sigma \times \tau$, represented by the elements $(i, j) \in B$, then the probability that some of these distances of the points of σ and τ is reduced by a factor greater than $e \cdot \beta \cdot |B|$ (the "bad" event) when projected to L can be bounded by the union bound inequality and by Lemma 5.1 for $\varphi = 1/(e\beta|B|)$ as:

$$\mathbf{Pr}\left[(\exists (i,j) \in B) : \frac{\delta'_{i,j}}{\delta_{i,j}} < \frac{1}{e\beta|B|}\right] \leq \sum_{(i,j)\in B} \mathbf{Pr}\left[\frac{\delta'_{i,j}}{\delta_{i,j}} < \frac{1}{e\beta|B|}\right] \leq \sum_{(i,j)\in B} \frac{e}{e\beta|B|} = \frac{1}{\beta}, \tag{5.13}$$

for any $\beta > 1$.

Since by Definition 5.4 any traversal T of σ and τ has a nonempty intersection with B, the discrete Fréchet distance of σ and τ has to be at least as big as the distance of some pair $(i, j) \in T \cap B$. These pairs of vertices have distance at least $d_{dF}(\sigma, \tau)/\theta$, and they are going to be reduced at most by the factor $e \cdot \beta \cdot |B|$ (with positive constant probability). The traversal T' of σ' and τ' that realizes $d_{dF}(\sigma', \tau')$ has to contain at least one of the pairs of B by Definition 5.4, since the pairs of the traversal T' are simultaneously the pairs of the traversal T of σ and τ (that contains the pairs of the vertices of σ and τ in the same order as the pairs of their projections in σ' and τ'). Thus $d_{dF}(\sigma', \tau') \geq d_{dF}(\sigma, \tau) / (e \cdot \beta \cdot \theta \cdot |B|)$, which proves the lemma. □

Intuitively we think of $\delta'_{i,j}$ as an approximation to $\delta_{i,j}$. Lemma 5.5 yields a naive $(\beta \cdot m^2)$-approximation for any $\beta > 1$ and $\theta = 1$. Let B be the set of all pairs $(i, j) \in \{1, \ldots, m\} \times \{1, \ldots, m\}$ such that $\|v_i - w_j\| = \delta_{i,j} \geq d_{dF}(\sigma, \tau)$. In the worst case B could

contain all m^2 pairs. Set B is a 1-guarding set. The correctness of the condition (i) of Definition 5.4 is provided directly by the definition of B. The condition (ii) follows by contradiction. If there would exist some traversal T such that $T \cap B = \emptyset$, then for all pairs $(i,j) \in T$ it would have to hold that $\|v_i - w_j\| < d_{dF}(\sigma,\tau)$. But then the traversal T would witness that $d_{dF}(\sigma,\tau) \leq \max_{(i,j)\in T} \|v_i - w_j\| < d_{dF}(\sigma,\tau)$, a contradiction.

Clearly, the approximation factor of Lemma 5.5 can be improved by the better choice of the set B. How can this be done is the question we explore in the following subsection.

5.3.2 Improved analysis for c-packed curves

In order to ensure that the number of the pairs of the indices that take part in the sum in the union bound inequality in (5.13) is not quadratic but at most linear in terms of the input size, we have to carefully select a small subset that satisfies the guarding set properties.

Building of the initial guarding set

We start with a simple construction of a θ-guarding set for any $\theta \geq 1$ by Algorithm 5. Lemma 5.6 proves that the resulting set is indeed a θ-guarding set.

Algorithm 5: Computing the θ-guarding set, $\theta \geq 1$

Data: $\delta = d_{dF}(\sigma,\tau)$, vertex-weighted graph $G = (V,E)$
Result: set B

$B \leftarrow \emptyset$
if $\delta_{1,1} \geq \delta/\theta$ **then**
  $B \leftarrow \{(1,1)\}$
else
  FIFO-Queue $\mathcal{Q} \leftarrow \{(1,1)\}$ /* Breadth-First-Search on $G = (V,E)$ */
  while $\mathcal{Q} \neq \emptyset$ **do**
    $(i,j) \leftarrow \text{pop}(\mathcal{Q})$
    foreach $((i,j),(i'j')) \in E$ **do**
      if $\delta_{i,j} < \delta/\theta$ and $\delta_{i',j'} < \delta/\theta$ **then**
        $\text{push}(\mathcal{Q},(i',j'))$
      else if $\delta_{i,j} < \delta/\theta$ and $\delta_{i',j'} \geq \delta/\theta$ **then**
        $B \leftarrow B \cup \{(i',j')\}$
return B

Lemma 5.6. *The set B obtained by Algorithm 5 is a θ-guarding set, for any $\theta \geq 1$.*

Proof. We have to show that the resulting set B satisfies the conditions of Definition 5.4. In the case that the distance $\delta_{1,1} \geq \delta/\theta$, it suffices to assign $B = \{(1,1)\}$, since any traversal of the curves σ and τ has to include the pair $(1,1)$. This is handled in lines 2-3 of Algorithm 5. For the rest of the proof let $\delta_{1,1} < \delta/\theta$.

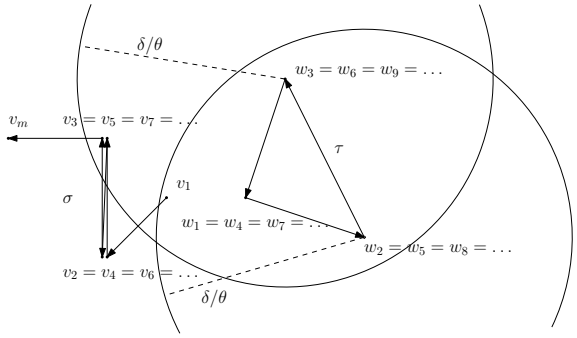
To see that the set B produced by Algorithm 5 satisfies the condition (i) of Definition 5.4, note that a pair (i',j') is added into B in line 12 only if $\delta_{i',j'} \geq \delta/\theta$, and such that the pair is reached by an edge from a pair (i,j) with $\delta_{i,j} < \delta/\theta$.

Algorithm 5 performs a Breadth-First-Search on $G = (V,E)$ (in lines 5-12), starting from $(1,1)$, that is initial content of the queue $\mathcal{Q}$. With each iteration of the BFS, the first element in $\mathcal{Q}$ is removed. Further pairs are pushed into $\mathcal{Q}$ only over the edges $((i,j),(i',j')) \in E$ with $\delta_{i,j} < \delta/\theta$ and $\delta_{i',j'} < \delta/\theta$. Since the graph G is directed and does not contain circuits, each edge in E can be explored by BFS at most once. The structure of G implies that each pair (vertex in V) can be added into $\mathcal{Q}$ at most three times. Thus, the while-loop of the BFS terminates with $\mathcal{Q} = \emptyset$.

We show the following invariant by induction over the steps of the BFS: after each iteration of the BFS, any traversal T contains either a pair (vertex) in B or a pair (vertex) in the queue $\mathcal{Q}$. The BFS starts in $(1,1) \in \mathcal{Q}$, with $\delta_{1,1} < \delta/\theta$. Since $(1,1) \in T$ for any T, the invariant initially holds. Let the invariant be satisfied for all vertices visited by the BFS before popping the pair (i,j) from $\mathcal{Q}$, where $\delta_{i,j} < \delta/\theta$ (since this had to hold when (i,j) was pushed into $\mathcal{Q}$). We observe the traversals whose pairs were not added into B yet (and thus have a pair in $\mathcal{Q}$). While processing the pair (i,j) (in lines 7-8), the traversal T may use one of the pairs $(i+1,j)$, $(i,j+1)$, or $(i+1,j+1)$ (connected to (i,j) by the edges of E). The next pair in T is either at distance less than δ/θ, thus is pushed into $\mathcal{Q}$ (lines 9-10), or at distance at least δ/θ, thus is added into B (lines 11-12). In both cases the invariant remains valid. The case distinction within the for-loop (line 8) is complete, since the pairs (i,j) with $\delta_{i,j} \geq \delta/\theta$ are never added into $\mathcal{Q}$.

The invariant is therefore valid at the end of the while-loop, when $\mathcal{Q} = \emptyset$, implying that the set B satisfies the condition (ii) of Definition 5.4. This closes the proof. □

Unfortunately, the set B built by Algorithm 5 can have a quadratic number of elements in terms of the input size, like the one in Figure 5.3 (marked with the outline). If the free-space matrix $F_{\delta/\theta}$ would have the "fork-like" structure for some $\theta \geq 1$, such that for every column j with $j \mod 3 = 1$ it holds for all pairs $\delta_{i,j} < \delta/\theta$ and thus $\phi_{i,j} = 1$ (except for $\delta_{m,j} \geq \delta/\theta$), and for every column j with $j \mod 3 = 2$ there are all pairs with $\delta_{i,j} \geq \delta/\theta$ and thus $\phi_{i,j} = 0$ (except for $\delta_{1,j} < \delta/\theta$). For the columns with $j \mod 3 = 0$ let $\phi_{1,j} = 1$, $\phi_{2,j} = 0$ and $\phi_{m,j} = 0$ (the rest may be filled arbitrarily). Then the set B built by Algorithm 5 would contain $(m-1) \cdot m/3 = O\left(m^2\right)$ entries. We note that this



$$F_{\delta/\theta} = \begin{pmatrix} \boxed{0} & \boxed{0} & 0 & \boxed{0} & \boxed{0} & 0 & \boxed{0} & \boxed{0} & \ldots \\ 1 & \boxed{0} & 1 & 1 & \boxed{0} & 1 & 1 & \boxed{0} & \ldots \\ \ldots & & & & & & & & \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \ldots \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & \ldots \\ 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 & \ldots \\ 1 & \boxed{0} & \boxed{0} & 1 & 0 & 0 & 1 & 0 & \ldots \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & \ldots \end{pmatrix}$$

Figure 5.3: The curves σ and τ (left) that yield a "fork-like" free-space matrix $F_{\delta/\theta}$ for some $\theta \geq 1$ (right). The pairs selected into B by Algorithm 5 are outlined.

cannot happen if the curves σ and τ are c-packed for some constant c, $c \geq 2$, as it will be discussed in the further text.

On the structure of the distance matrix

Lemma 5.7 states one property of the c-packed curves in $\mathbb{R}^d$, $d \geq 1$, which we apply afterwards in Lemma 5.8.

Lemma 5.7. *Given point v and a c-packed curve $\tau = w_1, \ldots, w_m$ from $\mathbb{R}^d$, then for any value $b > 0$ there exists a value $r \in [b/2, b]$, such that the $(d-1)$-sphere (the sphere in $\mathbb{R}^d$) centered at v with radius r intersects or is tangent to at most $2c$ edges of τ.*

Proof. Assume for the sake of contradiction that there exists $c' > 2c$, such that for any $r \in [b/2, b]$ there are at least c' edges of τ that intersect or are tangent to the $(d-1)$-sphere – the surface of the d-ball $\mathbf{B}(v, r)$. Let the *event points* be the points in $\mathbf{B}(v, b) \setminus \mathbf{B}(v, b/2)$, such that they are either

i) vertices w_i of τ or
ii) the points $w' \in \overline{w_i w_{i+1}}$, such that $\overline{vw'} \perp \overline{w_i w_{i+1}}$.

Let the set of events be $\mathcal{R} = \{R_1, \ldots, R_\ell\}$, and let $r_i = \|v - R_i\|$ for all $1 \leq i \leq \ell$. We may assume that the events R_i are sorted ascending by r_i. Let $r_0 = b/2$ and $r_{\ell+1} = b$, thus $r_0 \leq r_1 \leq \ldots \leq r_{\ell+1}$.

The number of the edges of τ that intersect or are tangent to the surface of $\mathbf{B}(v, r)$ is equal for all $r' \in [r_i, r_{i+1})$ and for all $0 \leq i \leq \ell$, since the number of such edges changes only in event points. By assumption there are at least c' edges of τ that intersect the

surface of $\mathbf{B}(v, r')$, for any $r' \in [r_i, r_{i+1})$ and for any $0 \leq i \leq \ell$. The length of the curve τ within $\mathbf{B}(v, b) \setminus \mathbf{B}(v, b/2)$ is

$$\sum_{i=0}^{\ell} \mathcal{L}\left(\tau \cap (\mathbf{B}(v, r_{i+1}) \setminus \mathbf{B}(v, r_i))\right) = \mathcal{L}\left(\tau \cap \left(\mathbf{B}(v, b) \setminus \mathbf{B}\left(v, \frac{b}{2}\right)\right)\right) \leq c \cdot b$$

since τ is c-packed. On the other side, it is

$$\sum_{i=0}^{\ell} \mathcal{L}\left(\tau \cap (\mathbf{B}(v, r_{i+1}) \setminus \mathbf{B}(v, r_i))\right) \geq \sum_{i=0}^{\ell} c' \cdot |r_{i+1} - r_i| = c' \cdot \left(b - \frac{b}{2}\right) > c \cdot b,$$

a contradiction. □

Lemma 5.8. *Given point v and a c-packed curve $\tau = w_1, \ldots, w_m$ from $\mathbb{R}^d$, and given a value $b > 0$, then for any pairwise disjoint set of intervals*

$$I \subseteq \{[i_1, i_2] \ : \ i_1, i_2 \in \mathbb{N} \wedge 1 \leq i_1 \leq i_2 \leq m\}$$

with $\|v - w_i\| \geq b$ for all indices $i \in [i_1, i_2] \cap \mathbb{N}$ where $[i_1, i_2] \in I$, there exists a value of $r \in [b/2, b]$ and a pairwise disjoint set of intervals

$$J \subseteq \{[j_1, j_2] \ : \ j_1, j_2 \in \mathbb{N} \wedge 1 \leq j_1 \leq j_2 \leq m\}$$

with the following properties:

(i) $|J| \leq c + 1$;

(ii) $(\forall\ [j_1, j_2] \in J)\ (\exists\ i_1 \leq i_2 < i_3 \leq i_4)\ :\ [i_1, i_2], [i_3, i_4] \in I \wedge j_1 = i_1 \wedge j_2 = i_4$;

(iii) $(\forall\ i \in [j_1, j_2] \cap \mathbb{N} : [j_1, j_2] \in J)\ :\ \|v - w_i\| \geq r$.

Proof. We set r to be the value of the same variable as in Lemma 5.7, $r \in [b/2, b]$, and we start with the given set I. Now we construct the set J by merging intervals of I as follows. Initially J is empty. We iterate over the intervals of I in the order of their starting points. Consider the first interval $[i_1, i_2]$ and the next interval in the order $[i_3, i_4]$. We merge them into one interval $[i_1, i_4]$ if there exists no vertex w_j with $i_2 < j < i_3$ such that $\|v - w_j\| < r$. We continue merging this interval with the intervals in I until we found a vertex w_j such that $\|v - w_j\| < r$. Then, we add the current merged interval to J and take the next interval from I and merge it with the proceeding intervals in the same manner. When there are no intervals left in I, we also add the current interval to J. Each time we add an interval to J (except possibly for the last one), we encountered two edges of τ that intersect the $(d-1)$-sphere of radius r centered at v. By Lemma 5.7 we have added

at most $c+1$ intervals to J (including the last interval). The other properties stated in the lemma follow by construction of J. See Figure 5.4 for an illustration of the merging process. □

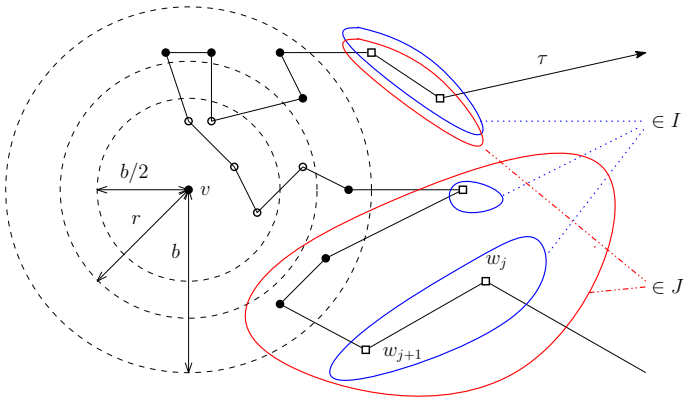


Figure 5.4: The process of Lemma 5.8 for the vertex v and the curve τ. The representation of the vertices of τ corresponds to their distance to v: squares for $\|v - w_j\| \geq b$, black circles for $b/2 \leq \|v - w_j\| < b$, white circles otherwise.

For a fixed vertex v of σ the intervals contained in the set I represent indices of the vertices w of the curve τ, such that the distance $\|v - w\|$ is at least $b = d_{dF}(\sigma, \tau)/\theta$, for some $\theta \geq 1$, thus the intervals I represent the entries 1 within a row/column of the free-space matrix F_b. Lemma 5.8 provides an algorithm with parameters: set of intervals I and a parameter b (or equivalently, parameter θ), such that the result of this algorithm is a set of intervals J that represent the entries 1 in the free-space matrix $F_{b/2}$. Before we can apply this algorithm to our guarding set B, we need to deal with some pairs whose presence in B is actually not necessary for keeping up the quality of the guarding set.

Avoidable pairs

If we have a θ-guarding set B obtained by Algorithm 5, we analyze if it is possible to remove some of the elements (i.e. pairs) of B while keeping the properties of the guarding set. For that sake we introduce the notion of an avoidable pair.

Definition 5.9 (Avoidable pair). *Let B be the θ-guarding set produced by Algorithm 5, and let $V = S_B \cup B \cup H_B$ be the partition of V implied by B. The pair $(i,j) \in B$ is called avoidable if there exist a pair $(i',j') \in B$ and two partial traversals T_1 and T_2 of σ and τ from $(1,1)$ to (i',j'), such that:*

(i) $\forall (i'',j'') \in (T_1 \cup T_2) \setminus \{(i',j')\}$ *it holds that* $(i'',j'') \in S_B$,

(ii) there exist pairs $(i,y_1) \in T_1$ *and* $(i,y_2) \in T_2$, *with* $y_1 < j < y_2$,

(iii) there exist pairs $(x_1, j) \in T_2$ *and* $(x_2, j) \in T_1$*, with* $x_1 < i < x_2$*.*

We say that the pair (i', j') and the partial traversals T_1 and T_2 from Definition 5.9 make the pair (i, j) avoidable.

We notice that for the pair to be avoidable, it suffices to have the conditions (i) and (ii), or (i) and (iii), since the remaining condition is implied by the monotonicity of the traversals. The definition of the avoidable pair (i, j) implies that any partial traversal of σ and τ from (i, j) to (m, m) has to have a nonempty intersection with $T_1 \cup T_2$.

Figure 5.5 shows the pairs selected by Algorithm 5 into the θ-guarding set B, for some $\theta \geq 1$, marked with polygonal red and blue bounds. The pairs within the red bound are avoidable, and the pairs within the blue bound are not. Two partial traversals T_1 and T_2 in S_B that make the red bounded pairs avoidable (as in Definition 5.9) are marked by arrows.

$$F_{\delta/\theta} = \begin{pmatrix} \cdots & & & & & & & \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & \ldots \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 & \ldots \\ 1 & 0 & 1 & 1 & 0 & 1 & 0 & \ldots \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & \ldots \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & \ldots \end{pmatrix}$$

Figure 5.5: Avoidable pairs from the θ-guarding set B (for some $\theta \geq 1$) are marked with red bound. Not avoidable pairs are marked with blue bound.

Lemma 5.10. *Given parameter* $\theta \geq 1$ *and the* θ*-guarding set* B*. Let* $B' \subseteq B$ *be the set of the avoidable pairs. Then* $B \setminus B'$ *is a* θ*-guarding set.*

Proof. The validity of the condition (i) of Definition 5.4 for the set $B \setminus B'$ is inherited from the set B. In order to prove the condition (ii), for the sake of contradiction, let there exist a traversal T of σ and τ such that $T \cap (B \setminus B') = \emptyset$. Since by Lemma 5.6 the traversal T of σ and τ satisfies $T \cap B \neq \emptyset$, there exists $(i, j) \in T \cap B'$, and we may assume that (i, j) is the last such avoidable pair along T. Let $(i', j') \in B \setminus B'$, T_1 and T_2 respectively be the pair in B and two traversals from Definition 5.9 that make the pair (i, j) avoidable.

We may assume that (i', j') is in $B \setminus B'$. To see this let $(i_1, j_1), (i_2, j_2), \ldots, (i_\ell, j_\ell)$ be the sequence of the pairs of indices, such that for all $x \in \{1, \ldots, \ell - 1\}$:

a) the pair $(i_x, j_x) \in B'$;

b) the pair (i_{x+1}, j_{x+1}) makes the pair (i_x, j_x) avoidable (from Definition 5.9); and

c) the pair $(i_\ell, j_\ell) \in B \setminus B'$.

Since it follows from Definition 5.9 and from the monotonicity of traversals that $i_1 < i_2 < \ldots \leq m$ and $j_1 < j_2 < \ldots \leq m$, such index ℓ has to exist. The partial traversals $T_1^{(\ell)}$ and $T_2^{(\ell)}$ from $(1,1)$ to (i_ℓ, j_ℓ) given by Definition 5.9, that make the pair $(i_{\ell-1}, j_{\ell-1})$ avoidable, satisfy the conditions of Definition 5.9 for the pair (i,j) as well. We assign $(i',j') = (i_\ell, j_\ell) \in B \setminus B'$, and thus it holds that $(i',j') \notin T$.

Let $(\hat{i},\hat{j})$ be the last pair along T such that $(\hat{i},\hat{j}) \in T \cap (T_1 \cup T_2)$ (there has to exist at least one such pair, w.l.o.g let it be in T_1). We construct the traversal T' of σ and τ out of the partial traversal of T_1 from $(1,1)$ to $(\hat{i},\hat{j})$ and the partial traversal of T from $(\hat{i},\hat{j})$ to (m,m). For the pairs $(i'',j'') \in T' \cap T_1$ it holds by Definition 5.9 that $(i'',j'') \in S_B$. Thus $(T' \cap T_1) \cap B = \emptyset$, since $B \cap S_B = \emptyset$.

Since $T \cap B = \emptyset$, it follows that $(T' \cap T) \cap B = \emptyset$. Therefore, for the traversal T' it holds that $T' \cap B = \emptyset$. This contradicts the assumption that B was the θ-guarding set, and proves that the condition (ii) of Definition 5.4 holds. Thus, $B \setminus B'$ is a θ-guarding set. □

Trimming the reachable area of a guarding set

Let B be a 1-guarding set for two curves σ and τ. We now want to modify B to shrink the number of pairs while maintaining the guarding property of Definition 5.4. It turns out that we can do this if we relax the approximation quality of the guarding set (which we denoted with θ). We perform this trimming as an algorithm in three phases:

(1) Remove all avoidable pairs from B.
(2) Trim the reachable area of B row by row.
(3) Trim the reachable area of B column by column.

In the following, we describe the trimming operation on a single row. Consider a vertex v_i of the curve σ and consider the intersection of B with the row of the distance matrix associated with v_i. Let I_i denote the set of intervals of the column indices that represent this intersection. We now apply Lemma 5.8 with parameter $b = d_{dF}(\sigma,\tau)$ to obtain a set of intervals J_i that can be used to trim the reachable area of B with respect to the ith row. Each interval in J_i covers a set of intervals of I_i. Let A_i be the subset of pairs of the ith row of which the column index is contained in an interval of J_i, but not contained in any interval of I_i. We call A_i the *filling pairs* of the row. We now want to trim the reachable area S_B defined by B along the vertices of the reachability graph which correspond to pairs of A_i. For this we will remove all vertices of B_i that are reachable from A_i and add the pairs of A_i to B. See Algorithm 6 for the pseudocode of this trimming operation. Figure 5.6 illustrates the process with an example.

The trimming operation for a single column is analogous to that on a single row, except that the initialization of the set I_i in the first line is now done as $I_i \leftarrow \{[j,j] : (j,i) \in B\}$

(the rows and the columns switch the places), and in the line 2 we use as parameters to the algorithm of Lemma 5.8 the set I_i and $b = d_{dF}(\sigma, \tau)/2$.

Algorithm 6: Trimming the reachable area for one row

Data: guarding set B, row index i, parameter $b > 0$
Result: modified guarding set B

1 $I_i \leftarrow \{[j, j] : (i, j) \in B\}$ /* each pair of B in the ith row produces an interval in I_i */
2 $J_i \leftarrow$ Algorithm of Lemma 5.8 using $I = I_i$ and $b = d_{dF}(\sigma, \tau)$ as parameters
3 $A_i := S_B \cap \left\{(i, j) : j \in \left(\bigcup_{[j_1,j_2]\in J_i} [j_1, j_2] \setminus \bigcup_{[i_1,i_2]\in I_i} [i_1, i_2]\right)\right\}$ /* compute filling pairs */
4 FIFO-Queue $\mathcal{Q} \leftarrow A_i$ /* find guarding pairs reachable from A_i via BFS */
5 **while** $\mathcal{Q} \neq \emptyset$ **do**
6 $(i, j) \leftarrow \text{pop}(\mathcal{Q})$
7 **foreach** $(i', j') \in \{(i + 1, j), (i + 1, j + 1)\}$ **do**
8 **if** $(i', j') \in B \setminus \mathcal{Q}$ **then**
9 $B \leftarrow B \setminus \{(i', j')\}$ /* remove the pair from B */
10 **else**
11 $\text{push}(\mathcal{Q}, (i', j'))$
12 $B \leftarrow B \cup A_i$ /* add pairs of A_i to B */
13 **return** B

$$F_b^{\text{before}} = \begin{pmatrix} \dots & & & & & \\ 0 & 0 & \boxed{0} & \boxed{0} & \boxed{0} & \dots \\ 0 & \boxed{0} & 1 & 1 & \boxed{0} & \dots \\ 0 & 1 & 1 & \boxed{0} & 0 & \dots \\ \boxed{0} & 1 & 1 & \boxed{0} & \boxed{0} & \dots \\ 1 & 1 & 1 & 1 & 1 & \dots \end{pmatrix} \qquad F_{b/2}^{\text{after}} = \begin{pmatrix} \dots & & & & & \\ 0 & 0 & \textcircled{0} & \textcircled{0} & \textcircled{0} & \dots \\ 0 & \textcircled{0} & 1 & 1 & \textcircled{0} & \dots \\ 0 & 1 & 1 & \textcircled{0} & 0 & \dots \\ \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \boxed{0} & \dots \\ 1 & 1 & 1 & 1 & 1 & \dots \end{pmatrix}$$

Figure 5.6: The elements of a guarding set (marked with squares) before (left) and after (right) applying of Algorithm 6 to the second row. The removed pairs are marked by circles

The following lemma shows the result of the trimming algorithm's phases, applied to the 1-guarding set B obtained by Algorithm 5.

Lemma 5.11. *Let B be a 1-guarding set.*

(i) After the first phase of the algorithm, which removes all avoidable pairs, the modified set B is a 1-guarding set.

(ii) After the second phase of the algorithm, which applies the trimming operation to each row with $b = d_{dF}(\sigma, \tau)$, the modified set B is a 2-guarding set.

(iii) After the third phase of the algorithm, which applies the trimming operation to each column with $b = d_{dF}(\sigma, \tau)/2$, the modified set B is a 4-guarding set.

Proof. The first part of the lemma follows directly from Lemma 5.10. We now prove the second part of the lemma statement. Condition (iii) of Lemma 5.8 ensures that any pair of a set A_i added to B corresponds to a pair of vertices $v \in \sigma$ and $w \in \tau$ with $\|v - w\| \geq b/2 = d_{dF}(\sigma, \tau)/2$. Indeed, the column indices of the pairs of A_i are contained in intervals of J_i. Therefore, after the second phase, the modified set B satisfies property (i) (distance property) in the definition of guarding sets if we set $\theta = 2$. Secondly, we argue that property (ii) (guarding property) is not invalidated after the trimming operation was applied to a row. Let B denote the guarding set before the trimming operation applied to the ith row and let B' denote the modifed guarding set after trimming. Clearly, the trimming operation does not add any avoidable pairs to B. Therefore we can assume that throughout the second phase no avoidable pairs are present.

Assume for the sake of contradiction that there exists a traversal T that contains a pair of B, but does *not* contain a pair of B'. Let (i', j') be the first pair along T that was removed from B during the trimming operation and let (i, j_2) be a pair of A_i that has a BFS-path to (i', j'). T must contain a pair (i, j_1) in the ith row and this pair cannot be contained in an interval of J_i (otherwise T would contain a pair of B'). Let T_1 be the partial traversal (path in G) of T that starts in $(1, 1)$, goes through (i, j_1), and ends in (i', j'). Since (i', j') was the first vertex along T in B, it follows that T_1 only visits vertices that are in S_B. Note that $i' > i$, since the BFS only visits row indices strictly greater than i. Since $A_i \subseteq S_B$, there must be a path T_2 in G from $(1, 1)$ through (i, j_2) to (i', j') that contains only vertices of S_B. Now, property (ii) of Lemma 5.8 implies that there must be a vertex (i, j'') in B, such that either $j_1 < j'' < j_2$ or $j_2 < j'' < j_1$. This implies that (i, j'') must be avoidable with respect to B. However, this contradicts the fact that B does not contain any avoidable pairs. This proves (ii). The third part of the lemma follows by a symmetric argument applied to the columns. □

5.3.3 Bounding the complexity of the modified guarding set

Given set B after the algorithm of Lemma 5.11. For every row of B (presented as matrix) let the pairwise disjoint set of intervals $R_i \subseteq \{[j_1, j_2] : j_1, j_2 \in \mathbb{N} \wedge 1 \leq j_1 \leq j_2 \leq m\}$ be a set of intervals on $\{1, \ldots, m\}$ of minimal size, such that for any $1 \leq j' \leq m$ there exist j_1 and j_2 with $j' \in [j_1, j_2] \in R_i$ if and only if $(i, j') \in B$. We can analogously define such pairwise disjoint sets C_j over the columns of B.

Lemma 5.8 implies that for every row i there is a set of pairwise disjoint intervals J_i constructed by line 2 of Algorithm 6, with $|J| \leq c+1$. Algorithm 6 takes into B only the pairs that belong to the subsets of the intervals of J_i that were in S_B as well. But since the pairs $(i,j) \in H_B$ such that $j \in [j_1, j_2] \in J_i$ have the property that any traversal using these pairs has to contain a pair in B prior to (i,j), we could have added such pairs too into B and then it would be $J_i = R_i$. Since we took only its subsets, it holds that for every $[j_1, j_2] \in R_i$ there is $[j_3, j_4] \in J_i$ with $j_3 \leq j_1 \leq j_2 \leq j_4$. By counting all intervals of R_i that are subset of one interval from J_i as one, we say that all such intervals R_i build one *extended group* of consecutive pairs within ith row. It follows that there are at most $c+1$ extended groups within i-th row. This process is repeated over columns as well. See Figure 5.7 for an illustration.

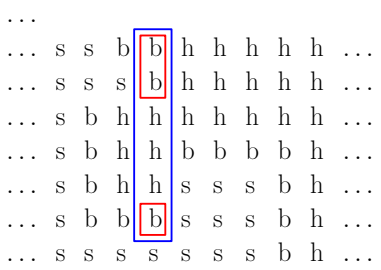


Figure 5.7: The pairs of the guarding set B (red) and its extended group (blue) within one column. The pairs denoted with s, b, and h are from S_B, B and H_B respectively

We have to note that the filling pairs added into B also imply the removal of a pair in B that lies in the same row but with higher column index. This does not necessarily apply for the last pair in the row. However, this can happen at most once per row, adding one pair (and one extended group) to the row. We obtain the following lemma.

Lemma 5.12. *In the guarding set produced by Algorithm 5 and modifed by the algorithm of Lemma 5.11, there are at most $c+1$ extended groups within a column, and $c+2$ extended groups within a row.*

To finally bound the complexity of our guarding set by Lemma 5.14, we show first Lemma 5.13.

Lemma 5.13. *For the guarding set produced by Algorithm 5 and after every phase of algorithm of Lemma 5.11 the following invariant holds: for every pair $(i,j) \in B$ there exists a pair $(i',j') \in S_B$ such that $((i',j'),(i,j)) \in E$.*

Proof. We call the pair (i', j') the predecessor pair. The construction of the guarding set B Algorithm 5 guarantees that a pair (i, j) is added into B if it is visited over an edge $((i', j'), (i, j)) \in E$, where $(i', j') \notin B$. Thus $((i', j') \in S_B$ as claimed.

The first phase of the algorithm of Lemma 5.11 removes the avoidable pairs from B, thus the invariant holds for the pairs that remain in B. The second phase runs Algorithm 6 upon a row and adds into B only pairs that were already in S_B, and that have also a predecessor in S_B. For every pair (i', j') which was in S_B before and is in H_B after Algorithm 6, it holds that the BFS passes (i', j') and then visits and subsequently removes the pairs from B. Therefore, the invariant remains valid for the pairs that remain in B. As for the pairs that were already in B their predecessors remain in S_B, so their status is not changed. The third phase is equivalent to the second one, and the invariant remains valid. □

Lemma 5.14. *The set B obtained by the algorithm of Lemma 5.11 is a 4-guarding set, containing at most $(3c + 4) \cdot m$ pairs.*

Proof. For every pair $(i, j) \in B$ one of the following holds true:

i) the index j is the smallest index of an extended group over the ith row;
ii) the index i is the smallest index of an extended group over the jth column;
iii) none of the above.

We argue that if neither i) nor ii) holds true, then it must be that $i - 1$ is the smallest index of an extended group over the jth column. Indeed, note that if neither i) nor ii) holds true, then $(i - 1, j)$ and $(i, j - 1)$ are part of an extended group and such groups can only contain pairs of B or H_B. Therefore, the pair $(i - 1, j - 1)$ must be in S_B because Lemma 5.13 implies that the pair (i, j) must have an ingoing edge from a pair in S_B. Now, since pairs of S_B and H_B cannot be directly connected by an edge of G, it must be that $(i - 1, j)$ and $(i, j - 1)$ are both in B. Thus, $i - 1$ is the smallest index of an extended group over the jth column.

We charge elements of B of type i) and of type ii) to their respective extended intervals. We charge elements of type iii) it to their extended interval over the column. Thus, extended intervals in the column are charged at most twice. By Lemma 5.12 we have at most $(c + 1)$ extended intervals per column and at most $(c + 2)$ extended intervals per row. This implies that altogether $|B| \leq (3c + 4) \cdot m$, as claimed. □

Lemma 5.5 and Lemma 5.14 imply the correctness of Theorem 5.15, that we state as conclusion of this section.

Theorem 5.15. *Given $c \geq 2$, for any two polygonal c-packed curves σ and τ of complexity m from $\mathbb{R}^d$, $d \in \{2,3,4,5,6,7\}$, and for any $\gamma \in (0,1)$ it holds that*

$$\mathbf{Pr}\left[\frac{d_{dF}(\sigma,\tau)}{d_{dF}(\sigma',\tau')} \leq e \cdot \frac{12c+16}{\gamma} \cdot m\right] \geq 1-\gamma,$$

where e is a constant, that equals 1 for $d \in \{2,3\}$, $1+2/\pi$ for $d \in \{4,5\}$, and $15/8$ for $d \in \{6,7\}$.

5.4 Lower bounds

Although one may see the upper bound of the previous section, being linear in the complexity of the input curves, as a weak one, in this section we show that the lower bound is as well linear in the complexity. We show that it may happen that for some two curves σ and τ it holds that the ratio between Fréchet distance of the curves and the Fréchet distance of the respective projection curves σ' and τ' is at least in $\Omega(m)$, where m is the complexity of the curves. This claim holds for both the discrete and the continuous version of the Fréchet distance, independently of the c-packedness of the curves σ and τ. We present an analogous claim for the dynamic time warping distance as well, but only in the case of the c-packed curves.

5.4.1 c-packed curves

We state Theorem 5.16 (for the discrete Fréchet distance), and then prove its correctness for the discrete and the continuous Fréchet distance, as well as for the dynamic time warping distance.

Theorem 5.16. *Given $c \geq 2$, there exist polygonal c-packed curves σ and τ of complexity m, such that for any $\gamma \in (0, 1/\pi)$*

$$\mathbf{Pr}\left[\frac{d_{dF}(\sigma,\tau)}{d_{dF}(\sigma',\tau')} \geq \frac{5\pi\gamma}{24} \cdot m\right] \geq 1-\gamma.$$

Proof of Theorem 5.16 for the discrete Fréchet distance. Let the curves σ and τ be from $\mathbb{R}^2$. Let $m = 2t+1$. Let the curve $\sigma = v_1, \ldots, v_{2t+1}$ be the line segment $\overline{v_1 v_{2t+1}}$, while the vertices $v_1, \ldots, v_{2t+1}$ are uniformly distributed on σ, i.e. $\|v_{i+1} - v_i\| = \|v_i - v_{i-1}\|$ for all $i \in \{2, \ldots, 2t\}$. Let $\tau = w_1, \ldots, w_{2t+1}$ be composed by two line segments $\overline{w_1 w_{t+1}}$ and $\overline{w_{t+1} w_{2t+1}}$, and the vertices $w_1, \ldots, w_{2t+1}$ are uniformly distributed on τ, i.e. $\|w_{j+1} - w_j\| = \|w_j - w_{j-1}\|$ for all $j \in \{2, \ldots, 2t\}$. Let $v_1 = w_1$ and $v_{2t+1} = w_{2t+1}$ and let $\angle w_{t+1} w_1 v_{2t+1} = \alpha$. An illustration is shown in Figure 5.8.

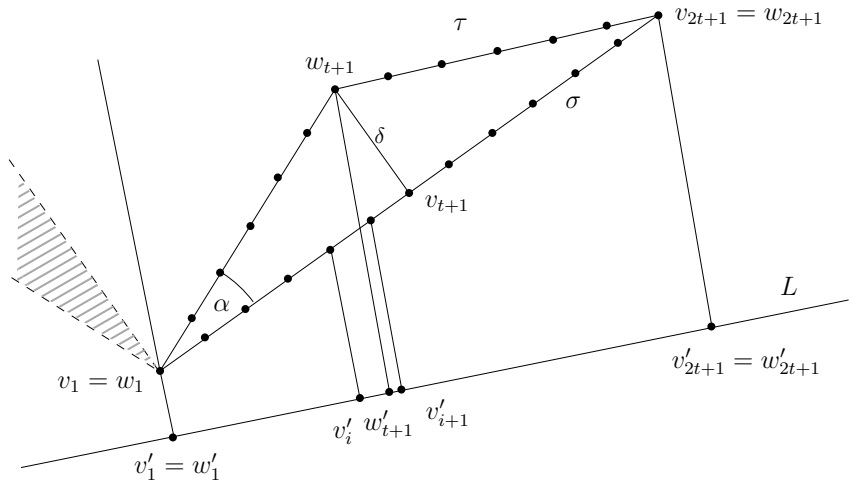


Figure 5.8: Curves that witness the lower bound for the discrete Fréchet distance case for c-packed curves (and for the continuous Fréchet distance and the dynamic time warping distance as well)

The curves σ and τ are c-packed for any constant $c \geq 2$. We may assume that $\mathcal{L}(\tau) = 2$. Then it holds that $\mathcal{L}(\sigma) = 2 \cdot |\cos\alpha|$, and for both discrete and continuous Fréchet distance, it holds that $\delta = d_{dF}(\sigma, \tau) = d_F(\sigma, \tau) = |\sin\alpha|$.

Let the straight line L support the unit vector $\mathbf{u}$, which is chosen uniformly at random on the unit sphere in $\mathbb{R}^2$, and let the curves σ and τ be projected onto L. Observe that the discrete Fréchet distance of σ and τ is realized by the pair $(t+1, t+1)$ in the traversal of σ and τ, thus $\|v_{t+1} - w_{t+1}\| = d_{dF}(\sigma, \tau) = \delta$. The vertex $w_{t+1} \in \tau$ is projected to w'_{t+1}, and w'_{t+1} lies either within σ' or outside of it.

If $w'_{t+1} \in \sigma'$, then $w'_{t+1} \in \overline{v'_i v'_{i+1}}$ for some $i \in \{1, \ldots, 2t\}$, i.e. w'_{t+1} is in one of the $2t$ line segments. We construct a traversal T' of σ' and τ' by induction. Let $(i, t+1), (i+1, t+1) \in T'$. For all indices $1 \leq j \leq t$, starting with $j = t$, we repeat the following until $(1, 1) \in T'$. Let x be the smallest index of a vertex v'_x added to T' (initially $x = i$). If $w'_j \in \overline{v'_x v'_{x+1}}$, we add the pair (x, j) to T'. Otherwise, it is $w'_j \in \overline{v'_{x-1} v'_x}$, and we add $(x-1, j)$ and (x, j) to T'. We conclude the iteration by decreasing j by one. We can analogously proceed the indices $t+2 \leq j \leq 2t+1$, until $(2t+1, 2t+1) \in T'$. By construction, T' is a traversal of σ' and τ'. Then, for each pair $(x, y) \in T'$ it is

$$\|w'_y - v'_x\| \leq \frac{\mathcal{L}(\sigma')}{2t} \leq \frac{\mathcal{L}(\sigma)}{2t} = \frac{|\cos\alpha|}{t}.$$

Therefore, it holds that

$$w'_{t+1} \in \sigma' \Rightarrow d_{dF}\big(\sigma', \tau'\big) \leq \frac{|\cos\alpha|}{t} \leq \frac{1}{t}.$$

For the event $w'_{t+1} \in \sigma'$ we bound the probability from below by

$$\mathbf{Pr}\left[w'_{t+1} \in \sigma'\right] \geq 1 - \alpha/\pi, \tag{5.14}$$

i.e. this event occurs if the perpendicular line to L is not parallel to some straight line laying in $\angle w_{t+1} w_1 v_{2t+1} = \alpha$ and including w_1 (tiled area in Figure 5.8). Then it holds that

$$\mathbf{Pr}\left[\frac{d_{dF}(\sigma,\tau)}{d_{dF}(\sigma',\tau')} \geq |\sin\alpha| \cdot t\right] \geq 1 - \frac{\alpha}{\pi}.$$

For $\alpha \in [0,1]$ it holds that $|\sin\alpha| \geq \alpha - \alpha^3/3! \geq 5\alpha/6$, thus for $\gamma = \alpha/\pi$ is for $\gamma \in (0, 1/\pi)$:

$$\mathbf{Pr}\left[\frac{d_{dF}(\sigma,\tau)}{d_{dF}(\sigma',\tau')} \geq \frac{5\pi\gamma}{6} \cdot t\right] \geq 1 - \gamma.$$

Since $t = (m-1)/2$ and $m \geq 2$, it follows that $t \geq m/4$. This proves the correctness of the theorem. □

In the continuous case, the linear factor is only m. We use the curves σ and τ from the discrete case.

Proof of Theorem 5.16 for the continuous Fréchet distance. For the case of the continuous Fréchet distance it holds that if $w'_{t+1} \in \sigma'$, then $\sigma' = \tau'$ and $d_F(\sigma',\tau') = 0$. Thus it holds that

$$\mathbf{Pr}\left[\frac{d_F(\sigma,\tau)}{d_F(\sigma',\tau')} \geq m\right] \geq \mathbf{Pr}\left[d_F\big(\sigma',\tau'\big) = 0\right] \geq 1 - \alpha/\pi$$

for any constant $\alpha \in (0,1)$. Thus, the continuous Fréchet distance will be reduced at least by a factor of m with probability at least $1-\gamma$, where $\gamma = \alpha/\pi$ and $\gamma \in (0, 1/\pi)$. □

The construction from the discrete Fréchet distance case can be extended to the dynamic time warping distance case, which we analyze next.

Proof of Theorem 5.16 for the dynamic time warping distance. For the curves σ and τ (defined in the discrete Fréchet distance case) it holds that

$$\begin{aligned} d_{DTW}(\sigma,\tau) &= \sum_{i=1}^{2t+1} \|v_i - w_i\| = 2 \cdot \left(\sum_{i=2}^{t} \|v_i - w_i\| \right) + \|v_{t+1} - w_{t+1}\| \\ &= 2 \cdot \left(\sum_{i=1}^{t} \|v_{i+1} - w_{i+1}\| \right) - \|v_{t+1} - w_{t+1}\| \\ &= 2 \cdot \left(\sum_{i=1}^{t} \frac{i \cdot |\sin\alpha|}{t} \right) - |\sin\alpha| = t \cdot |\sin\alpha| \end{aligned}$$

For the projection curves we extend the analysis for the discrete Fréchet distance case. Equation (5.14) states that $\mathbf{Pr}\left[w'_{t+1} \in \sigma'\right] \geq 1 - \alpha/\pi$. But if this event happens, then for all $1 \leq j \leq 2t+1$ is $w'_j \in \sigma'$, since $v_1 = w_1$ and $v_{2t+1} = w_{2t+1}$. For the rest of the proof we assume that this event happens.

Let $\mathcal{T}'$ be the set of all traversals of σ' and τ'. Let the set of the pairs $T' \in \mathcal{T}'$ be defined, such that for $1 \leq j \leq 2t+1$, the pair $(i,j) \in T'$ if and only if $\|v'_i - w'_j\|$ is minimal over all $1 \leq i \leq 2t+1$. Such set T' is a traversal of σ' and τ'. This is shown by induction, since $v_1 = w_1$ and $v_{2t+1} = w_{2t+1}$. Let the pair (i,j) be in T'. Then the closest vertex of σ' to the vertex w'_{j+1} has to be either v'_i or v'_{i+1}. The other vertices of σ' (either with smaller or greater index) cannot be the closest vertex to w'_{j+1} because of the order of the vertices on σ' and τ'. Thus, the pair $(i,j) \in T'$ is followed either by $(i+1, j+1)$ or $(i, j+1)$, and $T' \in \mathcal{T}'$ is a traversal. The possibility of $(i+1, j)$ is excluded, since we choose exactly one matched vertex for each j, $1 \leq j \leq 2t+1$.

Then it holds that

$$\begin{aligned} d_{DTW}\left(\sigma',\tau'\right) &= \min_{T \in \mathcal{T}'} \sum_{(i,j) \in T} \|v'_i - w'_j\| \leq \sum_{(i,j) \in T'} \|v'_i - w'_j\| \leq \frac{1}{2} \sum_{(i,j) \in T'} \|v'_i - v'_{i+1}\| \\ &\leq \frac{1}{2} \sum_{i=2}^{2t} \|v'_i - v'_{i+1}\| \leq \frac{1}{2} \cdot \mathcal{L}\left(\sigma'\right) \leq \frac{1}{2} \cdot \mathcal{L}\left(\sigma\right) = |\cos\alpha| \leq 1 \end{aligned}$$

with the probability $1 - \alpha/\pi$, and thus

$$\mathbf{Pr}\left[\frac{d_{DTW}(\sigma,\tau)}{d_{DTW}(\sigma',\tau')} \geq |\sin\alpha| \cdot t \right] \geq 1 - \frac{\alpha}{\pi}.$$

From this point on, we can repeat the final steps of the analysis of Theorem 5.16 for the discrete Fréchet distance, and obtain that the dynamic time warping distance will be reduced at least by a factor of $5\pi\gamma m/24$ with probability at least $1-\gamma$, for any $\gamma \in (0, 1/\pi)$, as claimed. □

5.4.2 General case curves

If the curves σ and τ are not c-packed, for any constant $c \geq 2$, then the ratio of the continuous Fréchet distances between σ and τ and their projection curves σ' and τ' can be at least linear in m, as claimed by Theorem 5.17. This event can happen with probability 1. We claim the same bound for the discrete Fréchet distance, by adapting the proof of the continuous case.

Theorem 5.17. *There exist the curves σ and τ of complexity m, such that if σ' and τ' respectively are their projections to the one-dimensional space that supports the unit vector chosen uniformly at random on the unit sphere in $\mathbb{R}^d$, then it holds that*

$$\frac{d_{dF}(\sigma, \tau)}{d_{dF}(\sigma', \tau')} \geq f(m),$$

where $f(m) \in \Omega(m)$.

Proof of Theorem 5.17 for the continuous Fréchet distance. We denote with ψ_k the star-like closed curve with $2k + 1$ vertices in $\mathbb{R}^2$, defined as $\psi_k = v_0, v_1, v_0, v_2, v_0, \ldots, v_k, v_0$. Let $v_i = (r_i, \vartheta_i)$ in polar coordinates be defined as $v_0 = (0, 0)$ and $v_i = (1, 2\pi \cdot (i-1)/k)$ for $1 \leq i \leq k$. Let $\sigma = \psi_k$ and $\tau = \psi_{k+1}$, and let k be even, $k \geq 6$. To have the same complexity for σ and τ, we can add two more points v_0 at the end of σ, thus the complexity is $m = 2k + 3$. We denote the indices of the curve τ with w_j, $0 \leq j \leq k + 2$. Figure 5.9 shows the curves σ and τ for $k = 12$ (in full blue and dotted red line, respectively).

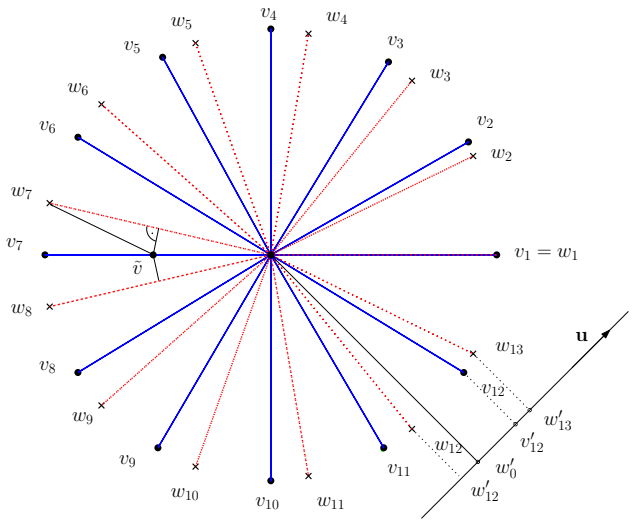


Figure 5.9: Two curves $\sigma = \psi_k$ (blue) and $\tau = \psi_{k+1}$ (red dotted), with parameter $k = 12$

The Fréchet distance between the curves σ and τ is (at most)

$$d_F(\sigma,\tau) = \frac{1}{2\cdot\cos(\pi/(k+1))}.$$

To show this, let M be the mapping of the points of σ and τ that witnesses the said distance. The curve τ has one more subcurve ("ray") of the star to be traversed. The "rays" v_0, v_1, v_0 and w_0, w_1, w_0 are equal, and they are mapped to each other by M at distance 0. The "rays" v_0, v_i, v_0 and w_0, w_i, w_0 for $1 \le i \le k/2$, and v_0, v_i, v_0 and w_0, w_{i+1}, w_0 for $k/2+2 \le i \le k$ are mapped pairwise by M at distance at most $2\pi/k - 2\pi/(k+1) \le \pi/(k+1)$. There remain two consecutive "rays" $w_0 w_{k/2+1} w_0$ and $w_0 w_{k/2+2} w_0$ that have to be mapped by the mapping M to $v_0 v_{k/2+1} v_0$. The point $\tilde{v}$ with coordinates $(1/(2\cdot\cos(\pi/(k+1))), \pi)$ is the intersection of a bisector of $\overline{w_0 w_{k/2+1}}$ with $\overline{v_0 v_{k/2+1}}$. Such point $\tilde{v}$ maps the subcurve of τ defined by the vertices $w_{k/2+1}, w_0, w_{k/2+2}$, thus the mapping M is completely described, and the Fréchet distance of σ and τ is at most the distance realized by M, that is $\|w_{k/2+1} - \tilde{v}\| = \|v_0 - \tilde{v}\| = 1/(2\cdot\cos(\pi/(k+1))) \ge \pi/(k+1)$, since $k \ge 6$. It holds that $d_F(\sigma,\tau) > 1/2$ for any $k \ge 2$.

We notice that between every two lines $\overline{w_0 w_j}$ and $\overline{w_0 w_{j+1}}$ there has to be one line $\overline{v_0 w_i}$, for $1 \le i, j \le k$. The opposite claim does not have to hold. Thus the distance between v_i and any of its neighboring w_j and w_{j+1} is at most $\max\{\|w_j - v_i\|, \|w_{j+1} - v_i\|\} \le \|w_{j+1} - w_j\| \le 2\pi/(k+1)$, since v_i is on the circular arc between w_j and w_{j+1}.

If we now project the curves σ and τ onto the straight line L that supports the unit vector $\mathbf{u}$, with $\mathbf{u}$ chosen uniformly at random on the unit sphere in $\mathbb{R}^2$, let $\sigma' = v'_0, v'_1, v'_0, \ldots, v'_k, v'_0$ and $\tau' = w'_0, w'_1, w'_0, \ldots, w'_{k+1}, w'_0$ be their projections respectively. The line $\overline{w_0 w'_0}$ satisfies one of the following two cases:

i) $\overline{w_0 w'_0} \| \overline{w_0 w_j}$ for some $1 \le j \le k+1$, or
ii) $\overline{w_0 w'_0}$ lies between the lines $\overline{w_0 w_j}$ and $\overline{w_0 w_{(j+1) \bmod (k+1)}}$ for some $1 \le j \le k+1$ (the modulo is added for the case that $j = k+1$).

Then in the first case, since k is even, the straight line $\overline{w_0 w'_0}$ lies between the lines $\overline{w_0 w_{(j+k/2) \bmod (k+1)}}$ and $\overline{w_0 w_{(j+k/2)+1 \bmod (k+1)}}$ (the lines through the two vertices on the opposite side of the star). For the simplicity of the notation, in the case $j = k/2+1$ the index $(j + k/2) \bmod (k+1)$ is replaced with $k+1$. Therefore, we may only consider the second case.

The projected curves σ' and τ' can be mapped to each other by a mapping M' as follows: let v_i be the vertex of σ that lies between $\overline{w_0 w_j}$ and $\overline{w_0 w_{j+1}}$ from the case definition (we omit the modulo for notation simplicity). Let v'_i be its projection. Then let the subcurves v'_0, v'_i, v'_0 and $w'_0, w'_j, w'_0, w'_{j+1}, w'_0$ be mapped to each other by mapping M'.

For the rest of the curves let $v'_0, v'_{i+\ell}, v'_0$ and $v'_0, v'_{i-\ell}, v'_0$ be matched to $w'_0, w'_{j+1+\ell}, w'_0$ and $w'_0, w'_{j-\ell}, w'_0$ respectively, as long as $i+\ell \leq k$ and $j+1+\ell \leq k+1$. Once this condition is not satisfied we replace $v'_{i+\ell}$ with $v'_{(i+\ell) \mod k}$, and $v'_{j+1+\ell}$ with $v'_{(j+1+\ell) \mod (k+1)}$ respectively. We map analogously $v'_0, v'_{i-\ell}, v'_0$ to $w'_0, w'_{j-\ell}, w'_0$ as long as $i-\ell \geq 1$ and $j-\ell \geq 1$. Note that each vertex w'_x of τ' (except w'_j and w'_{j+1}) is mapped to exactly one vertex of σ'.

Let $\hat{M}$ be the mapping of σ and τ such that the subcurves of σ and τ are mapped to each other if and only if their projections are mapped by M'

The value of $d_F(\sigma', \tau')$ is bounded from above by the maximum of the Fréchet distances between pairs of subcurves that are mapped by M'. The Fréchet distance between v'_0, v'_i, v'_0 and $w'_0, w'_j, w'_0, w'_{j+1}, w'_0$ is bounded from above by $\max\{\|v'_i - w'_j\|, \|v'_i - w'_{j+1}\|\} \leq \|w'_j - w'_{j+1}\| \leq 2\pi/(k+1)$. For the remaining pairs of subcurves, let the subcurves v'_0, v'_x, v'_0 and $w'_0, w'_{M'(x)}, w'_0$ be mapped to each other by M', for all $x \in [k] \setminus \{i\}$. The Fréchet distance between these two subcurves is $\|v'_x - w'_{M'(x)}\|$. By the construction of M' we conclude by induction that the points v_x of σ and $w_{\hat{M}(x)}$ of τ (whose projections are respectively v'_x and $w'_{M'(x)}$) have to lie both on a circular arc of length $2\pi/k$. Thus we have

$$\begin{aligned} d_F(\sigma', \tau') &\leq \max\{\|w'_j - w'_{j+1}\|, \max_{x \in [k]\setminus\{i\}} \{\|v'_x - w'_{M'(x)}\|\}\} \\ &\leq \max\{\frac{2\pi}{k+1}, \max_{x \in [k]\setminus\{i\}} \{\|v_x - w_{\hat{M}(x)}\|\}\} \leq \frac{2\pi}{k}. \end{aligned}$$

Therefore by projecting the curves σ and τ to any straight line the continuous Fréchet distance between the curves will be diminished at least by the factor

$$\frac{d_F(\sigma', \tau')}{d_F(\sigma, \tau)} < \frac{2\pi}{k} \cdot 2 = \frac{4\pi}{k}.$$

This yields the claimed linear lower bound, since $k = (m-3)/2$, and proves the theorem with $f(m) = (m-3)/(8\pi)$. □

Proof of Theorem 5.17 for the discrete Fréchet distance. The lower bound given by Theorem 5.17 holds for the discrete Fréchet distance as well, with $f(m) = (m-5)/(16\pi)$. We adapt the curves σ and τ from the proof for the continuous Fréchet distance as follows. Let us add to each "ray" v_0, v_i, v_0 of the curve ψ_k two vertices $\hat{v}_i$ (i.e. the "ray" becomes $v_0, \hat{v}_i, v_i, \hat{v}_i, v_0$), with polar coordinates $\hat{v}_i = (1/(2 \cdot \cos(\pi/(k+1)), \ 2 \cdot (i-1) \cdot \pi/k)$ (as for the vertex $\tilde{v}$ from the continuous case proof). The curve ψ_k contains now $4k+1$ vertices, and thus our curves σ and τ have complexity $m = 4k+5$. The rest of the construction and analysis can be used verbatim. □

5.5 Conclusion and open questions

Up to now we have not discussed if the analysis of the upper bounds presented in Section 5.3 can be extended to the continuous Fréchet distance or the dynamic time warping distance, as it was done in Section 5.4 for the lower bounds. While there is no free-space matrix for the DTW, and thus the analysis cannot be naturally extended for the DTW, in the continuous Fréchet distance case a free-space matrix would be replaced with a free-space diagram, and the notion of guarding set would include edges that bound the cells within the free-space diagram. Instead of avoidable pairs there would be avoidable areas, and our technique of Section 5.3 could be adapted.

However, the problem lies in the structure of the continuous Fréchet distance. The discrete Fréchet distance δ is always obtained as the distance between two vertices (so called "vertex-vertex event"). In the continuous Fréchet distance case the distance δ may be additionally realized by "vertex-edge events", where the Fréchet distance is realized between a vertex and a point on an edge, and by "vertex-vertex-edge events", where the Fréchet distance is realized between two vertices A and B on one curve, and a point on an edge of the other curve that lies on the bisector of the straight line $\overline{AB}$. For details confer the work of Driemel [64] and Driemel, Har-Peled and Wenk [66]. In these two cases a complementary claim to Lemma 5.1 is missing, in particular it is not known how to bound the ratio of the original and the projected distance, and subsequently to have the probabilities with such values that can be bounded jointly for the complete curves, as in Lemma 5.5 for the discrete case.

For the discrete Fréchet distance we showed that, in the worst case and under reasonable assumption for the input curves that they are c-packed for some constant $c > 0$, the distortion of the probabilistic embedding obtained from projecting to a randomly chosen line is at most linear in the complexity of the input curves. We showed that there are as well input curves that satisfy the same realistic assumptions and that witness the distortion that is at least linear in the complexity of the input curves. One may see this as a negative result, since we hoped that the Fréchet distance would be more robust under such embedding. However, we believe that this behavior occurs only for strongly conditioned curves, and not for a realistic input.

In the general case there exist polygonal curves that witness at least a linear distortion of the discrete Fréchet distance, in terms of the complexity of the input curves. It is an open question if the upper bound of the distortion can be matched with the lower bound in the general case.

6 Probabilistic smallest enclosing ball

6.1 Introduction

In the previous chapters the objective were polygonal curves. A polygonal curve describes the trajectory of some object by connecting the locations, i.e. the points in $\mathbb{R}^d$, in the order they are visited. Often, the locations visited by a chosen object are repeated, each with a respective frequency.

In this chapter we aim to gain knowledge from probabilistic points. If the set of possible locations of a point is given, each location accompanied with some probability, then a probabilistic point is intuitively defined by a discrete probability distribution, describing a set of possible locations in $\mathbb{R}^d$ where the point can appear at some moment.

Such setting is often met in practice. When looking for an optimal location for one (or more) mobile provider antenna(s) to serve N clients using cell phones, such that the maximal distance to a client is minimized, it is more realistic to observe the problem over probabilistic distributions of the locations of the clients, and ask for a good solution *in expectation*, than to optimize over all possible locations of all the clients. In addition, with some probability, a client may not be present at all. In the scenario that we optimize over all possible locations of all the clients, we would have an instance of some of the classical clustering problems, described in Section 2.4.

In particular, we are interested in the probabilistic smallest enclosing ball (pSEB) problem in high-dimensional Euclidean space. For this problem in a fixed-dimensional space, there is a fully polynomial time approximation scheme, provided by Munteanu, Sohler and Feldman [145]. However, their result assumes the dimension d of the ambient space to be a constant, and has an exponential running time dependency on d. To make the pSEB algorithm viable as a building block for high dimensional problems, as it is often the case in the machine learning context, it is desirable to reduce the dependence on the dimension from exponential to a small polynomial.

6.1.1 Problem definition

We consider a generalized median problem that we call the *set median problem*, and aim to solve it efficiently, in order to be able to use it as a building block for the other problems.

Definition 6.1 (**Set median problem**). *Let $\mathcal{P} = \{P_1, \ldots, P_N\}$ be a family of finite non-empty sets where for all $i \in [N]$, it is $P_i \subset \mathbb{R}^d$ and $n = \max\{|P_i| : i \in [N]\}$. The set median problem on $\mathcal{P}$ consists of finding a center $c \in \mathbb{R}^d$ that minimizes the cost function*

$$f(c) = \sum_{i=1}^{N} m(c, P_i),$$

where $m(c, P_i) = \max_{p \in P_i} \|c - p\|$.

The set median problem is a generalization to two well-known clustering problems. In case of singleton sets ($n = 1$), the set median problem is equivalent to the 1-median, defined in Equation (2.22) (also known as Fermat-Weber problem or geometric median). Also, if there is only one set ($N = 1$), the set median problem coincides with the smallest enclosing ball or 1-center problem, defined in Equation (2.21).

Next, we consider the probabilistic smallest enclosing ball (pSEB) problem, aiming to find a center that minimizes the expected maximum distance to points drawn from the input distributions. Let the input be a set $\mathcal{D} = \{D_1, \ldots, D_n\}$ of n discrete and independent probability distributions. The i-th distribution D_i is defined over a set of z possible locations $q_{i,j} \in \mathbb{R}^d \cup \{\perp\}$, for $j \in [z]$, where $\perp$ indicates that the i-th point is not present in a sampled set, i.e., $q_{i,j} = \perp \Leftrightarrow \{q_{i,j}\} = \emptyset$. We call the points, whose locations are given by the input distributions the **probabilistic points**. Each location $q_{i,j}$ is associated with the probability $p_{i,j}$, such that $\sum_{j=1}^{z} p_{i,j} = 1$, for every $i \in [n]$. Thus the probabilistic points can be considered as independent random variables X_i.

A **probabilistic set** X consisting of probabilistic points is also a random variable, where for each random choice of indices $(j_1, \ldots, j_n) \in [z]^n$ there is a realization $P_{(j_1,\ldots,j_n)} = X(j_1, \ldots, j_n) = (q_{1,j_1}, \ldots, q_{n,j_n})$. By independence of the distributions D_i, $i \in [n]$, it holds that

$$\mathbf{Pr}\left[X = P_{(j_1,\ldots,j_n)}\right] = \prod_{i=1}^{n} p_{i,j_i}.$$

The probabilistic smallest enclosing ball problem is defined as follows. Here we may assume that the distance from any point $c \in \mathbb{R}^d$ to the empty set is 0.

Definition 6.2 (**Probabilistic smallest enclosing ball problem**, cf. [145] Definition 2)**.** *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. The probabilistic smallest enclosing ball problem is to find a center $c^* \in \mathbb{R}^d$ that minimizes the expected smallest enclosing ball cost, i.e.,*

$$c^* \in \arg\min_{c \in \mathbb{R}^d} \mathbb{E}_X \left[m(c, X) \right],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$.

It was noted in [145] that the pSEB problem can be reduced to two different types of 1-median problems, and thus to two instances of the set median problem. We discuss this more in detail in Subsection 6.1.4.

The third problem we consider in this chapter is the support vector data description problem (SVDD). We introduced the Hilbert spaces and the kernel functions in Subsection 2.1.2. We formally state the SVDD problem next.

Definition 6.3 (**Support vector data description problem**)**.** *Given are an input set $P \subseteq \mathbb{R}^d$, and the kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ with implicit feature mapping $\phi \colon \mathbb{R}^d \to \mathcal{H}$, where $\mathcal{H}$ is an implicit Hilbert feature space. The task is to find*

$$c^* \in \arg\min_{c \in \mathcal{H}} \max_{p \in P} \|c - \phi(p)\| = \arg\min_{c \in \mathcal{H}} m(c, \phi(P)), \tag{6.1}$$

where $\phi(P) = \{\phi(p) \colon p \in P\}$.

It is known that the SVDD problem, originally introduced by Tax and Duin [166], is equivalent to the smallest enclosing ball problem in the feature space induced by the kernel function, what was shown by Tsang, Kwok and Cheung [169]. Note that the deterministic SVDD problem (Definition 6.3) is often stated with squared distances, as it is the case in the previous work [166, 169]. It does not matter whether we minimize the maximum distance or any of its powers or any other monotone transformation. In the probabilistic case this is not true. Consider, for instance, squared distances, i.e. if in Equation (6.1) there would be $m(c, \phi(P))^2$ instead of $m(c, \phi(P))$. Taking the expectation over the randomness of X, as we did in Definition 6.2, would yield the (weighted) sum of the squared distances, and thus, the resulting problem would be similar to a 1-means rather than a 1-median problem.

Huang *et al.* [106] have observed that minimizing the expected maximum squared distance corresponds to minimizing the expected area of an enclosing ball in $\mathbb{R}^2$. This observation can be generalized to the expected volume of an enclosing ball in $\mathbb{R}^p$ when the

p-th powers of distances are considered. Considering $p = 2$ might also have advantages when dealing with Gaussian input distributions due to their strong connection to squared Euclidean distances. In a general setting of the probabilistic smallest enclosing ball problem, however, it is natural to minimize in expectation the maximum Euclidean distance, since its radius is the primal variable to minimize. We proceed in such manner, to extend the SVDD to its probabilistic version.

The input is again a set $\mathcal{D}$ of n discrete and independent probability distributions, where $D_i \in \mathcal{D}$ is defined over a set of z locations $q_{i,j} \in \mathbb{R}^d \cup \{\perp\}$. Note that the mapping ϕ maps the locations $q_{i,j}$ from $\mathbb{R}^d$ to $\phi(q_{i,j})$ in $\mathcal{H}$, and we assume $\phi(\perp) = \perp$. Then the probabilistic SVDD problem is given by the following adaptation of Definition 6.2.

Definition 6.4. *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. Let $K\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel function with associated feature map $\phi\colon \mathbb{R}^d \to \mathcal{H}$. The probabilistic support vector data description (pSVDD) problem is to find a center $c^* \in \mathcal{H}$ that minimizes the expected SVDD cost, i.e.,*

$$c^* \in \arg\min_{c \in \mathcal{H}} \mathbb{E}_X \left[m(c, \phi(X)) \right],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$.

Throughout this chapter we work with *points* in $\mathbb{R}^d$. However, we intend to use the subgradient method, that utilizes *vectors* in $\mathbb{R}^d$ for translations. For each point $x \in \mathbb{R}^d$ we assign the vector $\mathbf{x} \in \mathbb{R}^d$ (from the origin to the point x). However, since for any two points $x, y \in \mathbb{R}^d$, and respective position vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$, it is $\|x - y\| = \|\mathbf{x} - \mathbf{y}\|$, we do not make the distinction between x and $\mathbf{x}$. It will always be clear from the context which object (point or vector) is meant at the moment.

6.1.2 Results in this chapter

In Section 6.2, we solve the set median problem (cf. Definition 6.1) on the collection of N deterministic point sets, using estimation and sampling techniques combined with a stochastic subgradient descent algorithm. The main result is presented by Theorem 6.15, with running time $O\left(\left(dn/\varepsilon^4\right) \cdot \log^2 1/\varepsilon\right)$.

The elements in the collection are sets of up to n points in $\mathbb{R}^d$. In Subsection 6.2.2, we discuss the possibility of further reducing the dependence on their size. In the previous work [145] the sets were summarized using strong coresets of size $1/\varepsilon^{\Theta(d)}$ for constant dimension d. This is not an option in high dimensions where, e.g. $d \approx n$. We show in Theorem 6.16 that no reduction below $\min\{n, \exp(d^{1/3})\}$ is possible unless one is willing

to sacrifice an additional approximation factor of roughly $\sqrt{2}$. However, we discuss the possibility to achieve roughly a factor $(\sqrt{2}+\varepsilon)$-approximation, both in off-line and in streaming setting, using well-known data structures.

In Subsection 6.3.1, we show how the $(1+\varepsilon)$-approximation algorithm for the set median problem improves the previously best FPTAS for the probabilistic smallest enclosing ball problem from $O\left(\left(dnz/\varepsilon^3\right)\cdot\log 1/\varepsilon + 1/\varepsilon^{O(d)}\right)$ to $O\left(\left(dnz/\varepsilon^4\right)\cdot\log^2 1/\varepsilon\right)$. In particular, the dependence on the dimension d is reduced from exponential to linear and, more notably, it occurs only in distance evaluations between points in d-dimensional Euclidean space, but not in the number of sampled points nor in the number of candidate centers to evaluate. This result is presented in Theorem 6.17.

The result of Subsection 6.3.1 enables working in very high D-dimensional Hilbert spaces, whose inner products and distances are given implicitly using positive semidefinite kernel functions, in Subsection 6.3.2. These functions can be evaluated in $O(d)$ time, although D is large or even unbounded depending on the kernel function. As an example, we extend the well-known support vector data description (SVDD) method to the probabilistic case. SVDD is equivalent to the smallest enclosing ball problem in the implicit high-dimensional feature space. The main result is presented in Theorem 6.19, and represent the first FPTAS for the probabilistic SVDD problem, with running time $O\left(dn\cdot\left(\left(z/\varepsilon^3\right)\cdot\log 1/\varepsilon + \left(1/\varepsilon^8\right)\cdot\log^2 1/\varepsilon\right)\right)$.

Throughout this chapter, we assume that the error parameter satisfies $0<\varepsilon<1/9$. All our results hold with constant probability, say $1/8$, which can be amplified to arbitrary $1-\eta$, $0<\eta<1$, by running $O(\log 1/\eta)$ independent repetitions and returning the minimum found.

6.1.3 Related work

We have discussed the deterministic clustering problems in Section 2.4. Here we present the related work on the *probabilistic* clustering problems.

The study of probabilistic clustering problems was initiated by Cormode and McGregor [59]. They developed approximation algorithms for the probabilistic settings of k-means, k-median as well as k-center clustering. For the probabilistic k-median in general metric space they gave a $(3+\varepsilon)$-approximation, while for the Euclidean k-median, as well for the both versions of k-means they produced a $(1+\varepsilon)$-approximation. The probabilistic metric k-center turns up to be the most challenging. Cormode and McGregor gave a $(1.582+\varepsilon)$-approximation, but with a blow-up on the number of clustering centers to $O\left((1/\varepsilon)\cdot\log^2 n\right)$, where n is the number of the probability distributions in the input, i.e. a bi-criteria $O(1)$-approximation. Even in the case that the distributions are reduced to

the choice between the appearance of the point on a single location or the point being absent, the blow-up on the centers remains, but the approximation factor becomes $1 + \varepsilon$. The running times of all algorithms of Cormode and McGregor is polynomial in n. Guha and Munagala [91] improved the result of Cormode and McGregor, by giving a polynomial time $O(1)$-approximation to the probabilistic k-center problem, that preserves the number of centers.

Munteanu, Sohler and Feldman [145] gave the first fully polynomial time $(1 + \varepsilon)$-approximation scheme (FPTAS) for the probabilistic Euclidean 1-center, i.e., the probabilistic smallest enclosing ball problem, in fixed dimensions, with running time of roughly $O(nd/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$. Since we build our pSEB algorithm in Section 6.3.1 upon their algorithm, we emphasize some of the aspects of the work from [145] in separate Subsection 6.1.4.

Huang, Li, Phillips and Wang [106] extended the notion of ε-kernels to the probabilistic points, but only for the single extent measure of the input set – the directional width. An ε-kernel is a subset of the input set, that approximates well the extent of the input. It yields a coreset for several extent measures, cf. the survey of Agarwal, Har-Peled and Varadarajan [8]. However, the ε-kernels have size exponential in d, i.e. $O\left(\varepsilon^{-O(d)}\right)$ [8].

Based on ε-kernels for probabilistic data of [106], Huang and Li [105] generalized the $(1 + \varepsilon)$-approximation of [145] to a polynomial time approximation scheme (PTAS) for Euclidean k-center in $\mathbb{R}^d$, for fixed constants k and d. The running time of the algorithm of Huang and Li [105] grows as a double exponential function of the dimension: $O\left(n^{O(1/\varepsilon)^{O(d)} \cdot d \cdot \mathrm{poly}\,(k)}\right)$. Even in our case, where $k = 1$, the running time is still prohibitive, i.e. exponential in $1/\varepsilon$, and doubly exponential in d. It is unclear how such a doubly exponential dependence on d could be reduced.

Huang and Li [105] gave a remark that to obtain a PTAS it is necessary to assume that k is a constant, as even the deterministic Euclidean k-center problem is **NP**-hard to approximate better than a factor of 1.822 for arbitrary k even in $\mathbb{R}^2$, as showed by Feder and Greene [80].

On probabilistic k-median clustering problem the only known result (other than [59]) was given by Lammersen, Schmidt and Sohler [128], who developed the first probabilistic coresets for uncertain datasets, by extending the technique of Chen [54]. Their coreset construction is aimed for the application in the streaming setting.

Kernel functions (introduced on page 21) simulate a Hilbert space in large or even unbounded dimensions but can be evaluated using simple low dimensional vector operations in the original dimension of input points [155, 158]. This enables simple spherical shape fitting using a smallest enclosing ball algorithm in the high dimensional feature space, which

implicitly defines a more complex and even non-convex shape in the original space. The smallest enclosing ball problem in kernel spaces was first observed by Tax and Duin [166]. A more subtle connection between (the dual formulations of) several kernel based methods in machine learning and the smallest enclosing ball problem was established by Tsang, Kwok and Cheung [169]. There are no known results on the probabilistic versions of the shape fitting and machine learning problems, prior to our work [125] presented in this chapter.

6.1.4 The best known probabilistic smallest enclosing ball algorithm

In this subsection we sketch the probabilistic smallest enclosing ball algorithm of Munteanu, Sohler and Feldman [145], and present some of their results. They assumed that the dimension of the ambient space $\mathbb{R}^d$ is a constant, thus the exponential dependence on d in the running time of their algorithm was not a problem.

We already mentioned that [145] showed a reduction of the probabilistic smallest enclosing ball problem to computing a solution for two *deterministic* instances of the set median problem. Their algorithm distinguishes between two cases.

The probability of obtaining a non-empty realization $P \neq \emptyset$ is small: Formally, in this case it is $\sum_{q_{i,j} \in Q} p_{i,j} \leq \varepsilon$, where $Q = \{q_{i,j} \colon q_{i,j} \neq \perp, i \in [n], j \in [z]\}$ (the set of non-empty locations). Then we have little chance of gaining information by sampling realizations. However, in [145] Lemma 6, it was shown that

$$(1-\varepsilon) \cdot \mathbb{E}_X \left[\sum_{p \in X} \|c-p\| \right] \leq \mathbb{E}_X \left[m(c,X) \right] \leq \mathbb{E}_X \left[\sum_{p \in X} \|c-p\| \right], \tag{6.2}$$

where $\mathbb{E}_X \left[\sum_{p \in X} \|c-p\| \right] = \sum_{i,j} p_{i,j} \cdot \|c - q_{i,j}\|$ is a weighted version of the deterministic 1-median problem (cf. Equation (2.22)). For the special case of the distributions that only differentiate between the point being present on a single location or not being present, Equation (6.2) was noted by Cormode and McGregor [59]. For arbitrary distributions, Equation (6.2) was noted by the authors of [145]. Thus, $\mathbb{E}_X \left[m(c,X) \right]$ is also a weighted instance of the set median problem, up to a factor of $(1-\varepsilon)$.

The probability that a realization contains at least one point is reasonably large: In this case, it is $\sum_{q_{i,j} \in Q} p_{i,j} > \varepsilon$, where $Q = \{q_{i,j} \colon q_{i,j} \neq \perp, i \in [n], j \in [z]\}$. By the definition of the expected value and $m(c, \emptyset) = 0$, we have

$$\mathbb{E}_X \left[m(c,X) \right] = \sum_{P \neq \emptyset} \mathbf{Pr} \left[X = P \right] \cdot m(c,P),$$

which is a weighted version of the set median problem with very large N (all possible point realizations).

We present their algorithm next, in order to be able to compare their findings to our result.

Algorithm 7: PSEB of Munteanu, Sohler and Feldman, cf. [145], Algorithm 1

Data: A set $\mathcal{D}$ of n point distributions over z locations in $\mathbb{R}^d$, a parameter $0 < \varepsilon < 1/2$
Result: A center $\hat{c} \in \mathbb{R}^d$

1 $Q \leftarrow \{q_{i,j} \colon q_{i,j} \neq \perp, i \in [n], j \in [z]\}$ /* the set of non-empty locations */
2 Set a sample size $k \in \Theta\left((d/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$
3 **if** $\sum_{q_{i,j} \in Q} p_{i,j} \leq \varepsilon$ **then**
4 - Pick a random sample R of k locations from $\mathcal{P} = Q$, where for every $r \in R$ we have $r = q_{ij}$ with probability proportional to p_{ij}
5 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1+\varepsilon)$-approximation to the 1-median of R, where the search for $\hat{c}$ is done over a grid of size $1/\varepsilon^{\Theta(d)}$
6 **else**
7 - Sample a set R of k non-empty realizations from the input distributions $\mathcal{D}$
8 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1+\varepsilon)$-approximation to the set median problem on R, where each realization is replaced by one strong coreset from [8] of size $O\left(1/\varepsilon^d\right)$. The search for $\hat{c}$ is done over a grid of size $1/\varepsilon^{\Theta(d)}$
9 **return** $\hat{c}$

Algorithm 7 runs in linear time in n, for sampling a constant number of realizations. Solving the subsampled problem takes only constant time, though exponential in the dimension. This yields a total running time of roughly $O(nzd/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)})$. Algorithm 7 can be extended into streaming settings using the exponential-sized strong coreset of Agarwal, Har-Peled and Varadarajan [8], that is already used in the line 8 of Algorithm 7. We state their main result as the following theorem.

Theorem 6.5 (cf. [145] Theorem 5)**.** *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. Let $\tilde{c} \in \mathbb{R}^d$ denote the output of Algorithm 7 on input $\mathcal{D}$, and let the approximation parameter be $0 < \varepsilon \leq 1/2$. Then, with constant probability, the output is a $(1+\varepsilon)$-approximation for the probabilistic smallest enclosing ball problem, i.e., it holds that*

$$\mathbb{E}_X\left[m(\tilde{c}, X)\right] \leq (1+\varepsilon) \min_{c \in \mathbb{R}^d} \mathbb{E}_X\left[m(c, X)\right].$$

The running time of Algorithm 7 is $O\left(\left(nzd/\varepsilon^3\right) \cdot \log^d(1/\varepsilon) + 1/\varepsilon^{O(d)}\right)$.

6.2 The set median problem

The set median problem, given in Definition 6.1, defines the distance $m(c, P)$ between a singleton set $\{c\}$ and some set P, $P \subset \mathbb{R}^d$, as the maximum distance $\max_{c,p \in P} \|c - P\|$. Extending this to a distance measure m between any two sets $A, B \subset \mathbb{R}^d$, defined by the maximum distance $m(A, B) = \max_{a \in A, b \in B} \|a - b\|$ does not yield a metric, since for any non-singleton set $C \subset \mathbb{R}^d$ it holds that $m(C, C) > 0$. However, throughout this chapter we consider only cases where, as in Definition 6.1, one of the sets $A = \{c\}$ is a singleton, and $B = P$ is an arbitrary non-empty set of points from $\mathbb{R}^d$. In order to directly apply results from the theory of metric spaces, we can define $m(A, B) = 0$ whenever $A = B$. Such an adaptation of m was done in the dissertation of Munteanu [143] (and later presented in [125]). For completeness we give here the proof that such m is a metric.

Lemma 6.6 (cf. [143] Lemma 5.2.3)**.** *Let $\mathcal{X}$ be the set of all finite non-empty subsets of $\mathbb{R}^d$. We define*

$$m(A, B) = \begin{cases} \max_{a \in A, b \in B} \|a - b\| & \text{if } A \neq B \\ 0 & \text{if } A = B \end{cases}$$

for any $A, B \in \mathcal{X}$. Then $(\mathcal{X}, m)$ is a metric space.

Proof. The non-negativity and symmetry properties of m follow from the corresponding metric properties in the Euclidean space $(\mathbb{R}^d, \|\cdot\|)$ and by definition. If $A = B$, then $m(A, B) = 0$ holds by definition. Otherwise there exist elements $a \in A$, $b \in B$, $a \neq b$, and thus $m(A, B) \geq \|a - b\| > 0$. This proves the identity of indiscernible elements.

To prove the validity of the triangle inequality, let $A, B, C \in \mathcal{X}$ be distinct. Let $a \in A$, $c \in C$ be points such that $m(A, C) = \|a - c\|$. For any $b \in B$, it holds that

$$m(A, C) = \|a - c\| \leq \|a - b\| + \|b - c\| \leq m(A, B) + m(B, C),$$

using triangle inequality in $(\mathbb{R}^d, \|\cdot\|)$ and the definition of m. Now consider the cases where at least two sets are equal. In case that $A = C$, the claim follows from the non-negativity property $m(A, C) = 0 \leq m(A, B) + m(B, C)$. In case that $A = B$, we have

$$m(A, C) = 0 + m(A, C) \leq m(A, B) + m(B, C).$$

The case $B = C$ is analogous, since $m(A, C) = m(A, C) + 0 \leq m(A, B) + m(B, C)$. □

To be able to apply the theory of convex analysis and optimization (cf. Section 2.2), we first note that our function f is a convex function. To see this, note that the Euclidean

norm is a convex function. Therefore the Euclidean distance to some fixed point is a convex function since every translation of a convex function is convex. The maximum of convex functions is a convex function and, finally, the sum of convex functions is again convex. We prove this claim for completeness.

Lemma 6.7. *The objective function f of the set median problem is convex.*

Proof. Let $x, y \in \mathbb{R}^d$, let $\lambda_1 \in [0,1], \lambda_2 = 1 - \lambda_1$, and let p_i^* maximize $\|\lambda_1 x + \lambda_2 y - p_i\|$ over all $p_i \in P_i$. Then, we have

$$\begin{aligned} f(\lambda_1 x + \lambda_2 y) &= \sum_{i=1}^{N} \max_{p_i \in P_i} \|\lambda_1 x + \lambda_2 y - p_i\| = \sum_{i=1}^{N} \left\| \lambda_1 x + \lambda_2 y - \underbrace{(\lambda_1 + \lambda_2)}_{=1} p_i^* \right\| \\ &= \sum_{i=1}^{N} \|\lambda_1 (x - p_i^*) + \lambda_2 (y - p_i^*)\| \leq \lambda_1 \sum_{i=1}^{N} \|x - p_i^*\| + \lambda_2 \sum_{i=1}^{N} \|y - p_i^*\| \\ &\leq \lambda_1 \sum_{i=1}^{N} \max_{p_i \in P_i} \|x - p_i\| + \lambda_2 \sum_{i=1}^{N} \max_{p_i \in P_i} \|y - p_i\| = \lambda_1 f(x) + \lambda_2 f(y), \end{aligned}$$

as claimed. □

In particular, the convexity of f implies that the subdifferential $\partial f(c)$ is non-empty for any center $c \in \mathbb{R}^d$ (cf. Lemma 2.17), and c is locally optimal if and only if $0 \in \partial f(c)$ (cf. Lemma 2.18). Moreover, any local optimum is also globally optimal by convexity (cf. Lemma 2.19). This implies that, if we find a $(1+\varepsilon)$-approximation to a local minimum of the convex function, the convexity implies that it is a $(1+\varepsilon)$-approximation to the global minimum as well.

Next preparatory step we make is to bound the Lipschitz constant of the function f by N.

Lemma 6.8. *The objective function f of the set median problem is N-Lipschitz continuous, i.e., it holds that $|f(x) - f(y)| \leq N \cdot \|x - y\|$ for all $x, y \in \mathbb{R}^d$.*

Proof. We fix any $x, y \in \mathbb{R}^d$. Let $p_i^* \in \operatorname{argmax}_{p_i \in P_i} \|x - p_i\|$. By the definition of f and applying the triangle inequality (of m) to every single term we have

$$\begin{aligned} |f(x) - f(y)| &= \left| \sum_{i=1}^{N} m(x, P_i) - \sum_{i=1}^{N} m(y, P_i) \right| \leq \sum_{i=1}^{N} |m(x, P_i) - m(y, P_i)| \\ &\leq \sum_{i=1}^{N} m(x, y) = N \cdot \|x - y\|. \end{aligned}$$

□

We may obtain a better bound if we limit the domain of f to a ball of small radius centered at the optimal solution, but the Lipschitz constant cannot be bounded by $o(N)$ in general. Namely, in the proof of Lemma 6.8, all inequalities become equalities if both x and y lie on the same half-axis, with $|x| > |y|$, and if all sets P_i, $i \in [N]$, are equal, each containing exactly one point, say 0. We will see in the next subsection how we can remove the dependence on N.

6.2.1 A subgradient descent method for the set median problem

We want to minimize f using the subgradient method, whose deterministic version we sketched in Section 2.2. We can (for now) say that our convex set of potential solutions is the whole set $\mathbb{R}^d$. Remember that we need to start from a chosen point $c_0 \in \mathbb{R}^d$. We need to choose the number of steps ℓ and the sequence of step sizes $\{h_i\}_{i=0}^{\ell}$. Finally, we need to know the value of R, that, in the deterministic model, was the distance from the starting point to an optimal solution, in order for the method to converge.

Computing a subgradient at a given point

We will use the fixed step size $h_i = s$, for all i. We choose the value of s, such that it fits our needs. However, to apply this method we must first *compute a subgradient* $g(c_i) \in \partial f(c_i)$ at the current center c_i. To this end we prove the following lemma, where we can define that $(c_i - p_j) / \|c_i - p_j\| = 0$, whenever $c_i = p_j$.

Lemma 6.9. *Let $c_i \in \mathbb{R}^d$ be any center. For each set $P_j \in \mathcal{P}$, let $p_j \in P_j$ be a point with $\|c_i - p_j\| = m(c_i, P_j)$. We have*

$$g(c_i) = \sum_{j=1}^{N} \frac{c_i - p_j}{\|c_i - p_j\|} \cdot \mathbb{1}_{c_i \neq p_j} \in \partial f(c_i), \tag{6.3}$$

i.e., $g(c_i)$ is a valid subgradient of f at c_i.

Proof. Let $c^* \in \operatorname{argmin}_{c \in \mathbb{R}^d} f(c)$. We first prove that for each term $j \in [N]$, it is

$$\left\langle \frac{c_i - p_j}{\|c_i - p_j\|} \cdot \mathbb{1}_{c_i \neq p_j}, c_i - c^* \right\rangle \geq m(c_i, P_j) - m(c^*, P_j). \tag{6.4}$$

Assume $c_i = p_j$, then $\langle 0, c_i - c^* \rangle = 0 \geq 0 - m(c^*, P_j) = m(c_i, P_j) - m(c^*, P_j)$. Otherwise, let $p_j^* \in P_j$ be a point such that $\left\| c^* - p_j^* \right\| = m(c^*, P_j)$. We have

$$\begin{aligned}
&\frac{\langle c_i - p_j, c_i - c^* \rangle}{\|c_i - p_j\|} = \frac{\langle c_i - p_j, c_i - p_j + p_j - c^* \rangle}{\|c_i - p_j\|} \\
= &\frac{\langle c_i - p_j, c_i - p_j - (c^* - p_j) \rangle}{\|c_i - p_j\|} = \frac{\langle c_i - p_j, c_i - p_j \rangle - \langle c_i - p_j, c^* - p_j \rangle}{\|c_i - p_j\|} \\
\overset{(2.4)}{\geq} &\frac{\|c_i - p_j\|^2 - \|c_i - p_j\| \cdot \|c^* - p_j\|}{\|c_i - p_j\|} \geq \|c_i - p_j\| - \left\| c^* - p_j^* \right\| = m(c_i, P_j) - m(c^*, P_j),
\end{aligned}$$

which follows by Cauchy-Schwarz inequality (Equation (2.4) in Lemma 2.4) and the choice of p_j^*, that gives $\|c^* - p_j\| \leq \left\| c^* - p_j^* \right\|$. Therefore, Equation (6.4) is satisfied.

Now summing Equation (6.4) over all $j \in [N]$ we have

$$\begin{aligned}
\left\langle \sum_{j=1}^{N} \frac{c_i - p_j}{\|c_i - p_j\|} \cdot \mathbb{1}_{c_i \neq p_j}, c_i - c^* \right\rangle &= \sum_{j=1}^{N} \left\langle \frac{c_i - p_j}{\|c_i - p_j\|} \cdot \mathbb{1}_{c_i \neq p_j}, c_i - c^* \right\rangle \\
&\overset{(6.4)}{\geq} \sum_{j=1}^{N} (m(c_i, P_j) - m(c^*, P_j)) = f(c_i) - f(c^*). \qquad \square
\end{aligned}$$

For brevity of presentation we omit the indicator function in any use of Lemma 6.9 in the remainder of this chapter.

The exact subgradient computation takes $O(dnN)$ time to calculate, since in each of the N terms of the sum we maximize over $|P_i| \leq n$ distances in d dimensions, to find a point in P_i that is farthest away from c. We are going to discuss the possibility of reducing the dependence on n later in Subsection 6.2.2. For now, we focus on removing the dependence on N. To this end we would like to replace the exact subgradient $g(c_i)$ by a uniform sample of only one non-zero term, which points into the right direction in expectation. In that case, a sampled subgradient could be computed in time $O(dn)$. We formalize this in the following lemma.

Lemma 6.10. *Let $c_i \in \mathbb{R}^d$ be any fixed center. For each set $P_j \in \mathcal{P}$, let $p_j \in P_j$ be a point with $\|c_i - p_j\| = m(c_i, P_j)$. Let $\tilde{g}(c_i)$ be a random vector, that takes the value $\tilde{g}(c_i) = (c_i - p_j) / \|c_i - p_j\|$ for $j \in [N]$ with probability $1/N$ each. Then, $\mathbb{E}\left[\|\tilde{g}(c_i)\|^2\right] \leq 1$ and $\mathbb{E}[\tilde{g}(c_i)] = g(c_i)/N$, where $g(c_i) \in \partial f(c_i)$ is the subgradient given in Lemma 6.9.*

Proof. The vector $\tilde{g}(c_i)$ is normalized by definition, except if an index j is chosen such that $c_i = p_j$, in which case $\|\tilde{g}(c_i)\| = 0$. Thus, $\mathbb{E}\left[\|\tilde{g}(c_i)\|^2\right] = \mathbb{E}\left[\|\tilde{g}(c_i)\|\right] \leq 1$ holds. We have as well that

$$\mathbb{E}\left[\tilde{g}(c_i)\right] = \sum_{j=1}^{N} \frac{1}{N} \cdot \frac{c_i - p_j}{\|c_i - p_j\|} = \frac{1}{N} \sum_{j=1}^{N} \frac{c_i - p_j}{\|c_i - p_j\|} \overset{(6.3)}{=} \frac{g(c_i)}{N},$$

using Lemma 6.9 (Equation (6.3)). □

Probabilistic subgradient descent algorithm

We can now adapt the deterministic subgradient method from [148], that we have sketched in Section 2.2, using the random unbiased subgradient of Lemma 6.10 in such way that the result is in expectation a $(1+\varepsilon)$-approximation to the optimal solution. This method is presented in Algorithm 8. Given an initial center c_0, a fixed step size s, and a number of iterations ℓ, Algorithm 8 iteratively picks a set $P_j \in \mathcal{P}$ uniformly at random and chooses a point $p_j \in P_j$ that attains the maximum distance to the current center. This point is used to compute an approximate subgradient using Lemma 6.10. The algorithm finally outputs the best center found in all iterations.

Algorithm 8: Stochastic subgradient method

Data: A family of non-empty sets $\mathcal{P} = \{P_1, \ldots, P_N\}$, where $P_i \subset \mathbb{R}^d$
Result: A center $\tilde{c} \in \mathbb{R}^d$

```
Determine an initial center c_0
Fix a step size s
Fix the number of iterations ℓ
for i ← 1 to ℓ do
    Choose an index j ∈ [N] uniformly at random, and compute g̃(c_{i−1}), cf.
     Lemma 6.10
    c_i ← c_{i−1} − s · g̃(c_{i−1})
return c̃ ∈ argmin_{c∈{c_i : i∈{0,...,ℓ}}} f(c)
```

The following theorem bounds in expectation the quality of the output that our subgradient algorithm returns. It is a probabilistic adaptation of Theorem 2.20, where the subgradient descent algorithm was deterministic.

Theorem 6.11. *Consider Algorithm 8 on input $\mathcal{P} = \{P_1, \ldots, P_N\}$ for the set median problem with objective function f, see Definition 6.1. Let an initial center c_0, a step size*

s, and a number of iterations ℓ in Algorithm 8 be chosen. Let $c^* \in \operatorname{argmin}_{c\in\mathbb{R}^d} f(c)$. Let $R = \|c_0 - c^*\|$. Then

$$\mathbb{E}_{\tilde{c}}\left[f(\tilde{c}) - f\left(c^*\right)\right] \leq N \cdot \frac{R^2 + (\ell+1)s^2}{2(\ell+1)s}, \tag{6.5}$$

where the expectation is taken over the random variable $\tilde{c} \in \operatorname{argmin}_{c\in\{c_i\colon i\in\{0,\ldots,\ell\}\}} f\left(c\right)$, i.e., the output of Algorithm 8.

Proof. Assume that we have reached a center $c_i \in \mathbb{R}^d$ while running Algorithm 8. Recall from Lemma 6.10 that $\mathbb{E}_{\tilde{g}}\left[\left\|\tilde{g}(c_i)\right\|^2 \mid c_i\right] \leq 1$ and $\mathbb{E}_{\tilde{g}}\left[\tilde{g}(c_i) \mid c_i\right] = g(c_i)/N$, since we assumed to compute a subgradient in c_i, and the expectation was taken over the randomness of $\tilde{g}$. We have

$$\begin{aligned}\mathbb{E}_{\tilde{g}}\left[\left\|c_{i+1} - c^*\right\|^2 \mid c_i\right] &= \mathbb{E}_{\tilde{g}}\left[\left\|c_i - s\tilde{g}(c_i) - c^*\right\|^2 \mid c_i\right] \\ &= \mathbb{E}_{\tilde{g}}\left[\left\|c_i - c^*\right\|^2 + \left\|s\tilde{g}(c_i)\right\|^2 - 2\left\langle s\tilde{g}(c_i), c_i - c^*\right\rangle \mid c_i\right]\end{aligned}$$

Using expected value over the choice of $\tilde{g}$, and the linearity of expectation, we further have

$$\begin{aligned}\mathbb{E}_{\tilde{g}}\left[\left\|c_{i+1} - c^*\right\|^2 \mid c_i\right] &\leq \mathbb{E}_{\tilde{g}}\left[\left\|c_i - c^*\right\|^2 \mid c_i\right] + s^2 - 2s\left\langle \mathbb{E}_{\tilde{g}}\left[\tilde{g}(c_i) \mid c_i\right], c_i - c^*\right\rangle \\ &= \mathbb{E}_{\tilde{g}}\left[\left\|c_i - c^*\right\|^2 \mid c_i\right] + s^2 - \frac{2s}{N}\left\langle g(c_i), c_i - c^*\right\rangle.\end{aligned}$$

The law of total expectation (cf. Lemma 2.23) implies, by taking expectations over c_i on both sides and rearranging, that

$$\mathbb{E}_{\tilde{g}}\left[\left\|c_i - c^*\right\|^2\right] + s^2 \geq \mathbb{E}_{\tilde{g}}\left[\left\|c_{i+1} - c^*\right\|^2\right] + \frac{2s}{N}\mathbb{E}_{c_i}\left[\left\langle g(c_i), c_i - c^*\right\rangle\right]. \tag{6.6}$$

We sum Equation (6.6) for all values of i, $i \in \{0, \ldots, \ell\}$, such that the terms $\mathbb{E}_{\tilde{g}}\left[\left\|c_k - c^*\right\|^2\right]$ for $k \in \{1, \ldots, \ell\}$ on both sides cancel. Since $\mathbb{E}_{\tilde{g}}\left[\left\|c_0 - c^*\right\|^2\right] = \left\|c_0 - c^*\right\|^2$, we obtain

$$\left\|c_0 - c^*\right\|^2 + (\ell+1)s^2 \geq \mathbb{E}_{\tilde{g}}\left[\left\|c_{\ell+1} - c^*\right\|^2\right] + \frac{2s}{N}\sum_{i=0}^{\ell}\mathbb{E}_{c_i}\left[\left\langle g(c_i), c_i - c^*\right\rangle\right]. \tag{6.7}$$

Note that for any set $\mathcal{Y}$ of positive real-valued random variables Y_i, we have $(\forall i\colon Y_i \in \mathcal{Y})$ $\min_{Y\in\mathcal{Y}}(Y) \leq Y_i$. This implies $(\forall i\colon Y_i \in \mathcal{Y})\, \mathbb{E}\left[\min_{Y\in\mathcal{Y}}(Y)\right] \leq \mathbb{E}\left[Y_i\right]$, and thus,

$$\mathbb{E}\left[\min_{Y\in\mathcal{Y}}(Y)\right] \leq \min_{Y\in\mathcal{Y}}\mathbb{E}\left[Y\right]. \tag{6.8}$$

Now, we can continue the derivation from Equation (6.7), using the subgradient property of $g(c_i)$ (cf. Equation (6.3)), Equation (6.8), and the fact that $\mathbb{E}_{\tilde{g}}\left[\|c_{\ell+1} - c^*\|^2\right] \geq 0$. We obtain

$$\begin{aligned}\|c_0 - c^*\|^2 + (\ell+1)s^2 &\geq \frac{2s}{N}\sum_{i=0}^{\ell}\mathbb{E}_{c_i}\left[\langle g(c_i), c_i - c^*\rangle\right] \overset{(6.3)}{\geq} \frac{2s}{N}\sum_{i=0}^{\ell}\mathbb{E}_{c_i}\left[f\left(c_i\right) - f\left(c^*\right)\right] \\ &\geq \frac{2s}{N}\left(\ell+1\right)\cdot \min_{i\in\{0,\ldots,\ell\}}\mathbb{E}_{c_i}\left[f\left(c_i\right) - f\left(c^*\right)\right] \\ &\overset{(6.8)}{\geq} \frac{2s(\ell+1)}{N}\cdot\mathbb{E}_{c_i}\left[\min_{i\in\{0,\ldots,\ell\}}\left(f\left(c_i\right) - f\left(c^*\right)\right)\right].\end{aligned}$$

The output of Algorithm 8 is $\tilde{c}$, which is the point in $\{c_i \colon i = 0, \ldots, \ell\}$ that minimizes $f\left(c_i\right)$. Rearranging the latter equation and substituting $R = \|c_0 - c^*\|$, yields

$$\mathbb{E}_{\tilde{c}}\left[f\left(\tilde{c}\right) - f\left(c^*\right)\right] = \mathbb{E}_{c_i}\left[\min_{i\in\{0,\ldots,\ell\}} f\left(c_i\right) - f\left(c^*\right)\right] \leq N\cdot\frac{R^2 + (\ell+1)s^2}{2(\ell+1)s}.$$

This closes the proof of the theorem. □

Our aim now is to choose the parameters ℓ, s, and c_0 of Algorithm 8 in such a way that the bound given in Theorem 6.11 becomes at most $\varepsilon f(c^*)$, thus giving a guarantee that the center $\tilde{c}$ we found by Algorithm 8 is a $(1+\varepsilon)$-approximation to the set median problem. We discuss separately how to do so for each of the parameters.

Choosing a starting point

Lemma 6.12 shows that we can choose the initial center c_0, and bound its initial distance $R = \|c_0 - c^*\|$ to the optimal solution proportional to the average cost $O\left(f(c^*)/N\right)$ with constant probability using a simple Markov argument.

Lemma 6.12. *Choose a set P from $\mathcal{P} = \{P_1, \ldots, P_N\}$ uniformly at random and let c_0 be an arbitrary point of P. Then, for any constant $0 < \gamma_1 < 1$ it holds that $R = \|c_0 - c^*\| \leq f(c^*)/(\gamma_1 N)$ with probability at least $1 - \gamma_1$.*

Proof. Define the random variable $X = m(c^*, P)$. We have $X \geq 0$ from Lemma 6.6. Its expectation equals

$$\mathbb{E}\left[X\right] = \sum_{i=1}^{N}\mathbf{Pr}\left[P = P_i\right]\cdot m(c^*, P_i) = \sum_{i=1}^{N}\frac{1}{N}\cdot m(c^*, P_i) = \frac{f(c^*)}{N}.$$

Thus, by Markov's inequality (cf. Lemma 2.26), we have

$$\mathbf{Pr}\left[X > \frac{f(c^*)}{\gamma_1 N}\right] \leq \gamma_1.$$

Now choose an arbitrary $c_0 \in P$. We have $R = \|c_0 - c^*\| \leq m(c^*, P) \leq f(c^*)/(\gamma_1 N)$ with probability at least $1 - \gamma_1$. □

Choosing the step size and the number of iterations

To guess properly the values of the step size s, and the number of iterations ℓ, we assume that we know the value of R. Then, we can set the step size to $s = R/\sqrt{\ell+1}$. Theorem 6.11 (Equation (6.5)) and Lemma 6.12 imply that for some constant C

$$\mathbb{E}_{\tilde{c}}\left[f(\tilde{c}) - f\left(c^*\right)\right] \overset{(6.5)}{\leq} \frac{NR}{\sqrt{\ell+1}} \leq \frac{N}{\sqrt{\ell+1}} \cdot \frac{f\left(c^*\right)}{\gamma_1 N} \leq \frac{C f(c^*)}{\sqrt{\ell+1}}$$

holds with constant probability. Thus, we only need to run the algorithm for $\ell \in O\left(1/\varepsilon^2\right)$ iterations to get $\mathbb{E}_{\tilde{c}}\left[f(\tilde{c}) - f\left(c^*\right)\right]$ within $\varepsilon f(c^*)$ error. But choosing this particular step size requires knowing the optimal center in advance. To get around this, we attempt to estimate the average cost. More formally, we are interested in a constant factor approximation of $f(c^*)/N$. It turns out that we can do this based on a small sample of the input sets, unless our initial center is already a good approximation. But in the latter case we do not care about all the subsequent steps or step sizes, since we have already found a good center after the initialization. The proof technique is originally from Kumar, Sabharwal and Sen [126] (cf. their Theorem 5.4), and is adapted here to work in our setting with sets of points.

Lemma 6.13. *There exists an algorithm that, based on a sample $\mathcal{S} \subseteq \mathcal{P}$ of size $|\mathcal{S}| = 1/\varepsilon$, returns an estimate $\tilde{R}$ and an initial center c_0 in time $O(dn/\varepsilon)$, such that, with constant probability, one of the following holds:*
a) $\varepsilon \cdot f(c^*)/N \leq \tilde{R} \leq \left(2/\varepsilon^3\right) \cdot f(c^*)/N$ *and* $\|c_0 - c^*\| \leq 8 \cdot f(c^*)/N$*;*
b) $\|c_0 - c^*\| \leq 4\varepsilon \cdot f(c^*)/N$*.*

Proof. Let $\Theta = f(c^*)/N$ be the value we want to approximate. Let c_0 be the initial center chosen as described in Lemma 6.12 with absolute constant $\gamma_1 = 1/8 \geq 1/81 > \varepsilon^2$. Consider the two balls $\mathbf{B}_1(c_0, \varepsilon\Theta)$ and $\mathbf{B}_2(c^*, \left(1/\varepsilon^2\right)\Theta)$. Then $\|c_0 - c^*\| \leq 8\Theta$ holds with constant probability $1 - \gamma_1$, and thus $c_0 \in \mathbf{B}_2$, since $8 < 1/\varepsilon^2$.

Let $\mathcal{Q}$ consist of all sets of $\mathcal{P}$ that are fully contained in $\mathbf{B}_2$. We have

$$|\mathcal{Q}| \geq (1 - \varepsilon^2)N, \tag{6.9}$$

since otherwise $f(c^*) \geq \sum_{P \in \mathcal{P} \setminus \mathcal{Q}} m(c^*, P) > \left(\varepsilon^2 N\right) \cdot \left(\Theta/\varepsilon^2\right) = f(c^*)$.

Now, sample a collection $\mathcal{S}$ of $1/\varepsilon$ sets, each uniformly from $\mathcal{P}$. All of our samples are completely contained in the ball $\mathbf{B}_2$ with constant probability. By a union bound inequality (cf. Lemma 2.25) over the elements of $\mathcal{S}$, the probability that this fails is at most $\gamma_2 \leq \varepsilon^2 \cdot 1/\varepsilon = \varepsilon < 1/8$.

Let $\tilde{R} = \sum_{P \in \mathcal{S}} m(c_0, P)$ be our estimate. We can compute $\tilde{R}$ in time $O\left(dn/\varepsilon\right)$, since for all $P \in \mathcal{S}$, it is $|P| \leq n$. We need to show that $\tilde{R}$ is close to the average cost Θ as we have claimed. To this end, consider the following two cases:

At most $(1-2\varepsilon)|\mathcal{Q}|$ sets of $\mathcal{Q}$ are completely contained in $\mathbf{B}_1$: In this case we have a constant probability $1 - \gamma_3$, for $\gamma_3 < 2\varepsilon < 2/8$, that there is a set $Q \in \mathcal{Q} \cap \mathcal{S}$ that contains some point $q \in Q$ that lies outside $\mathbf{B}_1$. Therefore, we have

$$\tilde{R} \geq m(c_0, Q) \geq \|c_0 - q\| \geq \varepsilon\Theta. \tag{6.10}$$

For the upper bound, note that the diameter of $\mathbf{B}_2$ is $2\Theta/\varepsilon^2$. Since all our samples are contained in that ball, we have

$$\tilde{R} = \sum_{P \in \mathcal{S}} m(c_0, P) \leq |\mathcal{S}| \cdot \frac{2\Theta}{\varepsilon^2} = \frac{2\Theta}{\varepsilon^3}. \tag{6.11}$$

Equations (6.10) and (6.11) imply that the claim a) holds.

At least $(1-2\varepsilon)|\mathcal{Q}|$ sets of $\mathcal{Q}$ are completely contained in $\mathbf{B}_1$: Suppose the second item b) does not hold, i.e. we have $R = \|c_0 - c^*\| > 4\varepsilon\Theta$. We can bound the number of sets that are not fully contained in $\mathbf{B}_1$ (thus, either not in $\mathcal{Q}$, or in $\mathcal{Q}$ but not contained in $\mathbf{B}_1$) by

$$|\mathcal{P} \setminus \mathcal{Q}| + 2\varepsilon|\mathcal{Q}| \overset{(6.9)}{\leq} \varepsilon^2 N + 2\varepsilon N \leq 3\varepsilon N. \tag{6.12}$$

Let $\mathcal{B} = \{P \in \mathcal{P} \colon P \subseteq \mathbf{B}_1\}$ be the remaining family of sets in $\mathcal{P}$ that are fully contained in $\mathbf{B}_1$. Equation (6.12) implies that $|\mathcal{B}| \geq (1 - 3\varepsilon)N$.

Now we compare the cost of using the center c_0 to the optimal cost. For every $P \in \mathcal{P}$ we have $|m(c^*, P) - m(c_0, P)| \leq \|c_0 - c^*\|$ by the triangle inequality. This inequality implies, that for each $P \in \mathcal{B}$ it holds that $m(c^*, P) - m(c_0, P) \geq (\|c_0 - c^*\| - \varepsilon\Theta) - m(c_0, P) \geq \|c_0 - c^*\| - 2\varepsilon\Theta$. So using the assumption $\|c_0 - c^*\| > 4\varepsilon\Theta$, and $0 < \varepsilon < 1/9$ we can deduce

$$
\begin{aligned}
f(c^*) - f(c_0) &= \sum_{P \in \mathcal{B}} (m(c^*, P) - m(c_0, P)) + \sum_{P \in \mathcal{P} \setminus \mathcal{B}} (m(c^*, P) - m(c_0, P)) \\
&\geq |\mathcal{B}|(\|c_0 - c^*\| - 2\varepsilon\Theta) - (N - |\mathcal{B}|)\,\|c_0 - c^*\| \\
&= 2|\mathcal{B}| \cdot (\|c_0 - c^*\| - \varepsilon\Theta) - N \cdot \|c_0 - c^*\| \\
&\geq 2N \cdot (1 - 3\varepsilon) \cdot (\|c_0 - c^*\| - \varepsilon\Theta) - N \cdot \|c_0 - c^*\| \\
&= N \cdot [\|c_0 - c^*\| \cdot (1 - 6\varepsilon) - \varepsilon\Theta \cdot (2 - 6\varepsilon)] \\
&> N \cdot [4\varepsilon\Theta \cdot (1 - 6\varepsilon) - \varepsilon\Theta \cdot (2 - 6\varepsilon)] = 2\varepsilon\Theta N\,(1 - 9\varepsilon) > 0.
\end{aligned}
$$

This contradicts the optimality of c^*. Thus, claim b) holds in this case. □

Lemma 6.13 has the following consequence. Either the initial center c_0 is already a $(1 + 4\varepsilon)$-approximation, in which case we are done, or we are close enough to an optimal solution, and we have a good estimate of the step size to find a $(1 + 4\varepsilon)$-approximation in a constant number of iterations. This will be formally stated in the proof of Theorem 6.15.

Deciding the best found approximation center

Another issue that we need to take care of, is finding the best center in the last line of Algorithm 8 efficiently. We cannot do this exactly since evaluating the cost even for one single center takes time $O(dnN)$. However, we can find a point that is a $(1 + \varepsilon)$-approximation of the best center in a finite set of candidate centers using a result from the theory of discrete metric spaces.

To this end we can apply our next theorem, which is originally from Indyk and Thorup. It was published in the dissertation of Indyk [110], as Theorem 31, and in the paper by Thorup in [168], as Theorem 34. We adapt it here to work in our setting. The main difference is that in the original work the set of input points and the set of candidate solutions are identical. In our setting, however, we have that the collection of input sets and the set of candidate solutions may be completely distinct, and our distance measure is the modified maximum distance, shown to be a metric in Lemma 6.6.

Theorem 6.14. *Let $\mathcal{Q}$ be a set of uniform samples with repetition from $\mathcal{P}$. Let $\mathcal{C}$ be a set of candidate solutions. Let $a \in \mathcal{C}$ minimize $\sum_{Q \in \mathcal{Q}} m(a, Q)$, and let $\hat{c} = \operatorname{argmin}_{c \in \mathcal{C}} f(c)$. Then*

$$
\mathbf{Pr}\left[\sum_{P \in \mathcal{P}} m\,(a, P) > (1 + \varepsilon) \sum_{P \in \mathcal{P}} m\,(\hat{c}, P)\right] \leq |\mathcal{C}| \cdot e^{-\varepsilon^2 |\mathcal{Q}|/64}. \tag{6.13}
$$

Proof. Let b be an arbitrary center in $\mathcal{C}$ with

$$\sum_{P\in\mathcal{P}} m(b,P) > (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P). \tag{6.14}$$

If there is no such center then all centers are good approximations, in which case the theorem is trivial. There are at most $|\mathcal{C}|$ choices for b. We study the random variable

$$X = \sum_{Q\in\mathcal{Q}} \frac{m(b,Q) - m(\hat{c},Q) + m(\hat{c},b)}{2m(\hat{c},b)} = \sum_{Q\in\mathcal{Q}} h(Q), \tag{6.15}$$

where $m(\hat{c},b) = \|\hat{c}-b\|$, and $h(Q)$ denote the summands of Equation (6.15). Since m is a metric, by triangle inequality it holds that X is the sum of random variables between 0 and 1. The *bad* event is $X \le |\mathcal{Q}|/2$.

If we denote by $\mathbb{1}_{Q\in\mathcal{Q}}$ the indicator function that $Q\in\mathcal{Q}$, then we have $X = \sum_{Q\in\mathcal{Q}} h(Q) = \sum_{Q\in\mathcal{P}} h(Q)\cdot\mathbb{1}_{Q\in\mathcal{Q}}$. It holds that

$$\mathbb{E}[X] = \sum_{Q\in\mathcal{P}} \mathbb{E}[h(Q)\cdot\mathbb{1}_{Q\in\mathcal{Q}}] = \sum_{Q\in\mathcal{P}} \left(\mathbb{1}_{Q\in\mathcal{Q}}\cdot\frac{|\mathcal{Q}|}{|\mathcal{P}|}\cdot h(Q)\right) = \frac{|\mathcal{Q}|}{|\mathcal{P}|}\sum_{Q\in\mathcal{P}} h(Q). \tag{6.16}$$

Equation (6.14) and the triangle inequality for any set $Q\in\mathcal{P}$: $m(\hat{c},Q)+m(b,Q)\ge m(\hat{c},b)$, imply that

$$\begin{aligned}(2+\varepsilon)\sum_{Q\in\mathcal{P}} m(b,Q) &= \sum_{Q\in\mathcal{P}} m(b,Q) + (1+\varepsilon)\sum_{Q\in\mathcal{P}} m(b,Q)\\ &\overset{(6.14)}{>} (1+\varepsilon)\sum_{Q\in\mathcal{P}} m(\hat{c},Q) + (1+\varepsilon)\sum_{Q\in\mathcal{P}} m(b,Q) \ge (1+\varepsilon)\sum_{Q\in\mathcal{P}} m(\hat{c},b).\end{aligned}$$

Furthemore, using Equation (6.14) once again, it holds that,

$$\sum_{Q\in\mathcal{P}} (m(b,Q) - m(\hat{c},Q)) \overset{(6.14)}{>} \frac{\varepsilon}{1+\varepsilon}\sum_{Q\in\mathcal{P}} m(b,Q) > \frac{\varepsilon}{2+\varepsilon}\sum_{Q\in\mathcal{P}} m(\hat{c},b). \tag{6.17}$$

It follows that

$$\begin{aligned}\mathbb{E}[X] &\overset{(6.16)}{=} \frac{|\mathcal{Q}|}{|\mathcal{P}|}\sum_{Q\in\mathcal{P}} h(Q) \overset{(6.15)}{=} \frac{|\mathcal{Q}|}{|\mathcal{P}|}\sum_{Q\in\mathcal{P}}\left(\frac{m(b,Q)-m(\hat{c},Q)}{2m(\hat{c},b)} + \frac{1}{2}\right)\\ &\overset{(6.17)}{>} \frac{|\mathcal{Q}|}{2|\mathcal{P}|}\sum_{Q\in\mathcal{P}}\left(\frac{\varepsilon}{2+\varepsilon}+1\right),\end{aligned}$$

and the *bad* event is bounded by

$$\frac{|\mathcal{Q}|}{2} < \mathbb{E}[X] \cdot \frac{2+\varepsilon}{2+2\varepsilon} = \mathbb{E}[X] \cdot \left(1 - \frac{\varepsilon}{2+2\varepsilon}\right) = \mathbb{E}[X] \cdot (1-\eta), \tag{6.18}$$

where $\eta = \varepsilon/(2+2\varepsilon)$, and $\eta \geq \varepsilon/4$ holds since $\varepsilon < 1$. Using the Chernoff bound (cf. Lemma 2.27) we have that

$$\mathbf{Pr}\left[X \leq \frac{|\mathcal{Q}|}{2}\right] \overset{(6.18)}{\leq} \mathbf{Pr}[X < (1-\eta) \cdot \mathbb{E}[X]] \overset{(2.20)}{<} e^{-\eta^2 \mathbb{E}[X]/2} \leq e^{-\varepsilon^2|\mathcal{Q}|/64}, \tag{6.19}$$

since $\eta \geq \varepsilon/4$ and $\eta^2/(1-\eta) \geq \varepsilon^2/16$. Transforming X in Equation (6.19) we have

$$\mathbf{Pr}\left[\sum_{Q\in\mathcal{Q}} m(b,Q) \leq \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)\right] < e^{-\varepsilon^2|\mathcal{Q}|/64}. \tag{6.20}$$

Defining the set of *bad centers* $\mathcal{B} = \{b \in \mathcal{C} \colon \sum_{P\in\mathcal{P}} m(b,P) > (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P)\}$, we finally have, using the union bound inequality over all bad centers, that

$$\begin{aligned}
&\mathbf{Pr}\left[\forall b \in \mathcal{C}, \sum_{P\in\mathcal{P}} m(b,P) > (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P) \colon \sum_{Q\in\mathcal{Q}} m(b,Q) > \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)\right] \\
&= 1 - \mathbf{Pr}\left[\exists b \in \mathcal{C}, \sum_{P\in\mathcal{P}} m(b,P) > (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P) \colon \sum_{Q\in\mathcal{Q}} m(b,Q) \leq \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)\right] \\
&\geq 1 - \sum_{b\in\mathcal{B}} \mathbf{Pr}\left[\sum_{Q\in\mathcal{Q}} m(b,Q) \leq \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)\right] \overset{(6.20)}{\geq} 1 - |\mathcal{B}| \cdot e^{-\varepsilon^2|\mathcal{Q}|/64} \geq 1 - |\mathcal{C}| \cdot e^{-\varepsilon^2|\mathcal{Q}|/64}.
\end{aligned}$$

Then it holds in particular for $a \in \mathcal{C}$, that with probability at least $1 - |\mathcal{C}|\exp(-\varepsilon^2|\mathcal{Q}|/64)$, it is

$$\left(\sum_{P\in\mathcal{P}} m(a,P) > (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P)\right) \Rightarrow \left(\sum_{Q\in\mathcal{Q}} m(a,Q) > \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)\right). \tag{6.21}$$

But it holds that $\sum_{Q\in\mathcal{Q}} m(a,Q) \leq \sum_{Q\in\mathcal{Q}} m(\hat{c},Q)$ by optimality of a, so the contrapositive of Equation (6.21) yields that

$$\mathbf{Pr}\left[\sum_{P\in\mathcal{P}} m(a,P) \leq (1+\varepsilon)\sum_{P\in\mathcal{P}} m(\hat{c},P)\right] \geq 1 - |\mathcal{C}| \cdot e^{-\varepsilon^2|\mathcal{Q}|/64}.$$

This closes the proof of the theorem. □

A $(1+\varepsilon)$-approximation to the set median

Putting all pieces we developed in this section together, we obtain the following theorem, that is the main result on the set median problem – a $(1+\varepsilon)$-approximation algorithm to the set median problem.

Theorem 6.15. *Consider an input $\mathcal{P} = \{P_1, \ldots, P_N\}$, where for every $i \in [N]$ we have $P_i \subset \mathbb{R}^d$ and $n = \max\{|P_i| : i \in [N]\}$. There exists an algorithm that computes a center $\tilde{c}$ that is with constant probability a $(1+\varepsilon)$-approximation to the optimal solution c^* of the set median problem. Its running time is $O\left((dn/\varepsilon^4) \cdot \log^2(1/\varepsilon)\right)$.*

Proof. Set $\ell = (68/\varepsilon)^2$. Using Lemma 6.13, we distinguish the two possible cases.

(i) If our initial center c_0 satisfies $\|c_0 - c^*\| \leq 4\varepsilon f(c^*)/N$, then Lemma 6.8 yields

$$f(c_0) \leq f(c^*) + N\|c_0 - c^*\| \leq (1+4\varepsilon) f(c^*),$$

so the starting point is already a good center.

(ii) Otherwise, we have $\varepsilon \cdot f(c^*)/N \leq \tilde{R} \leq \left(2/\varepsilon^3\right) \cdot f(c^*)/N$ and $R = \|c_0 - c^*\| \leq 8 \cdot f(c^*)/N$ by Lemma 6.13. Thus, $\varepsilon^3 \tilde{R}/2 \leq f(c^*)/N \leq \tilde{R}/\varepsilon$. To improve this, we run the main loop of Algorithm 8 for the step sizes $s = \tilde{R}_j/\sqrt{\ell+1}$, where $\tilde{R}_j = 2^{j-1} \cdot \varepsilon^3 \tilde{R}$ for all values of $0 \leq j \leq \lceil \log(2/\varepsilon^4) \rceil$. For some particular value of j we have a 2-approximation given by

$$\frac{f(c^*)}{N} \leq \tilde{R}_j \leq 2 \cdot \frac{f(c^*)}{N}. \tag{6.22}$$

In this particular run, setting the step size $s = \tilde{R}_j/\sqrt{\ell+1}$, and incorporating it into the bound given in Theorem 6.11 (Equation (6.5)), we have that

$$\begin{aligned}\mathbb{E}_{\tilde{c}}\left[f(\tilde{c}) - f\left(c^*\right)\right] &\overset{(6.5)}{\leq} N \cdot \frac{R^2 + (\ell+1)s^2}{2(\ell+1)s} \leq N \cdot \frac{R^2 + \tilde{R}_j^2}{2\sqrt{\ell+1}\tilde{R}_j} \overset{(6.22)}{\leq} \frac{8^2+2^2}{2\sqrt{\ell+1}} f(c^*) \\ &\leq \frac{\varepsilon}{2} f(c^*).\end{aligned}$$

Using Markov's inequality (cf. Lemma 2.26) we have that

$$\mathbf{Pr}\left[f(\tilde{c}) - f(c^*) \geq 4\varepsilon f(c^*)\right] \leq \frac{\varepsilon f(c^*)}{2} \cdot \frac{1}{4\varepsilon f(c^*)} = \frac{1}{8} = \gamma_4.$$

The best center collected in all repetitions cannot be worse than this particular $\tilde{c}$ or c_0, see the cases of Lemma 6.13.

Finally we have a collection $\mathcal{C}$ of $|\mathcal{C}| \in O\left((1/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$ centers, and want to find one of them that is a $(1+\varepsilon)$-approximation for the best center in $\mathcal{C}$ using Theorem 6.14. We sample a collection of $(64/\varepsilon^2) \cdot \ln(8|\mathcal{C}|) \in O\left((1/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$ point sets from $\mathcal{P}$, and find the best center for this subset of points, which is the final output of our algorithm. By Theorem 6.14 this center is within another factor of $(1+\varepsilon)$ to the best in $\mathcal{C}$ with failure probability at most $\gamma_5 \leq 1/8$. The total approximation factor is thus at most $(1+4\varepsilon)(1+\varepsilon) \leq 1+9\varepsilon$. Rescaling ε yields the correctness. The total failure probability is at most $\gamma = \sum_{i=1}^{5} \gamma_i \leq 6/8$ by the union bound inequality, over all bad events in Lemma 6.13 and this theorem.

It remains to prove the running time of the algorithm. The initial center c_0 and the estimate $\tilde{R}$ can be computed in $O(dn/\varepsilon)$ time, see Lemma 6.13. The main loop of Algorithm 8 takes $O(dn)$ in each iteration, and runs for $\ell \in O\left(1/\varepsilon^2\right)$ iterations for a fixed step size. But we try $O(\log(1/\varepsilon))$ different step sizes. This makes up a running time of $O\left((dn/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$. Finally, we evaluate the objective function for $O\left((1/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$ centers for the sample of $O\left((1/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$ sets taken using Theorem 6.14. This can be done in time $O\left((dn/\varepsilon^4) \cdot \log^2(1/\varepsilon)\right)$, which dominates the running time of the whole algorithm. This closes the proof of the theorem. □

6.2.2 Reducing the dependence on the size of the input sets

To read the whole input we need the linear dependence on the size n of the input set. But, is it possible to perform afterwards the maximum distance computations on some smaller sketch set, whose size would be sublinear in n? If yes, we could explore the extension of the work of Munteanu, Sohler and Feldman [145] into the streaming setting, where after reading the input once in linear time, the subsequent computations were sublinear, even independent of n in the running time. To this end, a grid-based strong coreset of size $1/\varepsilon^{\Theta(d)}$ was used. However, here we focus on reducing the dependence on d, and an exponential dependence is not an option if we want to work in high dimensions. It turns out that, without introducing an exponential dependence on d, we would have to lose a constant approximation factor.

Pagh *et al.* [149] showed that any data structure that stores a subset of the input, and that approximates farthest neighbor queries to within less than a factor $\sqrt{2}$, must consist of $\min\{n, \exp(\Omega(d))\}$ points. Previously, Agarwal and Sharathkumar [9] carefully constructed an input point set of $\Omega\left(\exp(d^{1/3})\right)$ points in $\mathbb{R}^d$, to prove the lower bounds on streaming algorithms for several extent problems. Among the problems they considered is approximate farthest neighbor, with the lower bound on approximation factor of roughly $\sqrt{2}$.

In Theorem 6.16, we review the techniques of Agarwal and Sharathkumar [9] to show a slightly stronger result, namely: no small data structure can exist for answering maximum distance queries within a factor of less than roughly $\sqrt{2}$. In comparison to the cited results, Theorem 6.16 is not limited to the streaming setting (as in [9]), and it is not restricted to subsets of the input (as it is the case in [149]).

In Theorem 6.16, we reduce from the indexing problem, that is, Alice is given a vector $a \in \{0,1\}^n$ and a random coin. She sends a single message to Bob. Bob has an index $i \in [n]$ and a random coin. Bob has to, based on the message from Alice, guess the value of the i-th bit of a, denoted a_i, with probability at least $2/3$. The indexing problem is known to have $\Theta(n)$ one-way randomized communication complexity, i.e. the length of any message from Alice that helps Bob to guess a_i is linear in n. This result is stated in the paper of Kremer, Nisan and Ron [124], as Theorem 3.7.

Theorem 6.16. *Any data structure that, with probability at least $2/3$, α-approximates maximum distance queries on a set $S \subset \mathbb{R}^d$ of size $|S| = n$, for $\alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$, requires $\Omega\left(\min\{n, \exp\left(d^{1/3}\right)\}\right)$ bits of storage.*

Proof. We show that if there would be a value $\alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$, such that an α-approximation to the maximum distance query would be possible while keeping less than $\Omega\left(\min\{n, \exp\left(d^{1/3}\right)\}\right)$ bits of storage, then we would be able to solve the indexing problem with less than $\Omega(n)$ communicated bits, a contradiction.

It is known from [9] that there is a centrally symmetric point set K of size $\Omega\left(\exp\left(d^{1/3}\right)\right)$ on the unit hypersphere in $\mathbb{R}^d$ centered at the origin, such that for any pair of distinct points $p, q \in K$ it holds that

$$\sqrt{2}\left(1 - 2/d^{1/3}\right) \leq \|p - q\| \leq \sqrt{2}\left(1 + 2/d^{1/3}\right) \tag{6.23}$$

unless $p \neq -q$, in which case $\|p - q\| = 2$.

Let d be the smallest integer such that $d \geq 8$ and $n \leq \exp\left(d^{1/3}\right)$. We choose a set of $\exp\left(d^{1/3}\right)$ pairs of centrally symmetric points of K. We may assume that there is a lexicographic order of these pairs, so there is a mapping between the indices of a and the pairs of points of K, that is known to both Alice and Bob. Alice constructs the set S by including the first point of the i-th pair, denoted p_i, if and only if $a_i = 1$. She builds a data structure Σ_S which she sends to Bob.

Let the data structure be such that for any $x \in \mathbb{R}^d$, the answer to a query $\Sigma_S(x)$ satisfies

$$\frac{m(x,S)}{\alpha} \leq \Sigma_S(x) \leq m(x,S),$$

for some constant $1 < \alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$. We consider two cases:

(i) If $a_i = 1$, then p_i is included in S. Thus $\Sigma_S(-p_i) \geq m(-p_i, S)/\alpha = 2/\alpha$. Since $\alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$ and $d \geq 8$, it holds that

$$\Sigma_S(-p_i) \geq \sqrt{2} \cdot \frac{1 + 2/d^{1/3}}{1 - 4/d^{2/3}} > \sqrt{2}\left(1 + 2/d^{1/3}\right).$$

(ii) if $a_i = 0$, then $p_i \notin S$, and thus, $\Sigma_S(-p_i) \leq m(-p_i, S) \overset{(6.23)}{\leq} \sqrt{2}\left(1 + 2/d^{1/3}\right)$.

Therefore, if $\alpha < \sqrt{2}\left(1 - 2/d^{1/3}\right)$ Bob could, based on Σ_S, solve the indexing problem by querying $q_i = -p_i$. Consequently, any encoding of Σ_S uses $\Omega\left(\min\{n, \exp\left(d^{1/3}\right)\}\right)$ bits of space. □

On the positive side, it was shown by Goel, Indyk and Varadarajan [87], that a $\sqrt{2}$-approximate farthest neighbor to the point $c \in \mathbb{R}^d$ can always be found on the surface of the smallest enclosing ball of each set P_i, using linear preprocessing time $\tilde{O}(dn)$ and $\tilde{O}(d^2)$ query time. Thus, if we plug in the (weak) coresets of Bădoiu and Clarkson [20, 21] for the 1-center clustering problem (cf. Definition 2.29), of size $O(1/\varepsilon)$ instead of the entire sets $P_i \in \mathcal{P}$ to evaluate $m(c, P_i)$, we would have a sublinear time algorithm (in n) after reading the input, using a $\sqrt{2}(1 + \varepsilon)$-approximation to any query $m(c, P_i)$.

In a streaming setting the same $\sqrt{2}(1 + \varepsilon)$-approximation to any query $m(c, P_i)$ can be achieved using the *blurred-ball-cover* of Agarwal and Sharathkumar [9]. They defined a blurred-ball-cover to be (intuitively) a sequence of subsets K_i of the input set, each of the size $O(1/\varepsilon)$, such that the radii of the smallest enclosing balls of K_i are increasing. Each of these balls covers its predecessors, and all balls together cover the input set. Their data structure has size $O\left((d/\varepsilon^3) \cdot \log(1/\varepsilon)\right)$, and answers a farthest neighbor query in time $O\left((d/\varepsilon^3) \cdot \log(1/\varepsilon)\right)$.

There exists a $(1+\varepsilon)$-approximation to the farthest neighbor query, but at the cost of the running time. Goel, Indyk and Varadarajan [87] have shown that using $O\left(dn^{1+1/(1+\varepsilon)}\right)$ preprocessing time and $\tilde{O}(dn^{1/(1+\varepsilon)})$ query time, one can obtain a $(1 + \varepsilon)$-approximation for the farthest neighbor problem. Note that in this case, the preprocessing time is already superlinear in n, and in particular, the exponent is already larger than 1.7 for $1 + \varepsilon < \sqrt{2}$.

6.3 Applications of the set median problem

6.3.1 Probabilistic smallest enclosing ball

In this section, we apply our result on the set median problem (Theorem 6.15) to the probabilistic smallest enclosing ball problem. We adapt the framework of Algorithm 7, and obtain Algorithm 9. Depending on the two cases we discussed in Subsection 6.1.4, we sample a number of elements, non-empty locations or non-empty realizations, and solve the resulting set median problem using the samples in Theorem 6.15 for computing the approximate subgradients.

Algorithm 9 differs from Algorithm 7 mainly in three points. First, the number of samples in line 2 of Algorithm 7 had a dependence on d, that was hidden in the O-notation in the algorithm published in [145]. This is not the case in Algorithm 9. Second, the sampled realizations are not sketched using coresets of size $O\left(1/\varepsilon^d\right)$ any more (used in line 8 of Algorithm 7). Third, the running time of the actual optimization task (in lines 5 and 8 of Algorithm 7) is reduced using Theorem 6.15 instead of an exhaustive grid search over $1/\varepsilon^{\Theta(d)}$ grid points.

Algorithm 9: Probabilistic smallest enclosing ball

Data: A set $\mathcal{D}$ of n point distributions over z locations in $\mathbb{R}^d$, a parameter $0 < \varepsilon < 1/9$

Result: A center $\hat{c} \in \mathbb{R}^d$

1 $Q \leftarrow \{q_{i,j} : q_{i,j} \neq \perp, i \in [n], j \in [z]\}$ /* the set of non-empty locations */

2 Set a sample size $k \in O\left((1/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$

3 **if** $\sum_{q_{i,j} \in Q} p_{i,j} \leq \varepsilon$ **then**

4 - Pick a random sample R of k locations from $\mathcal{P} = Q$, where for every $r \in R$ we have $r = q_{ij}$ with probability proportional to p_{ij}

5 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1+\varepsilon)$-approximation using the sampled points R one-by-one for computing the approximate subgradients in the algorithm of Theorem 6.15

6 **else**

7 - Sample a set R of k non-empty realizations from the input distributions $\mathcal{D}$

8 - Compute $\hat{c} \in \mathbb{R}^d$ that is a $(1+\varepsilon)$-approximation using the sampled realizations R one-by-one for computing the approximate subgradients in the algorithm of Theorem 6.15

9 **return** $\hat{c}$

In all previously discussed problems in this thesis, where the sampling was required, the uniform random sampling sufficed and was used. The samples in Algorithm 9 are picked with probabilities proportional to $p_{i,j}$, and thus, are not necessarily pairwise equal.

The solution for this issue is provided by using a weighted reservoir sampler, described by Efraimidis [74], and based on the previous work of Chao [52]. It enables sampling of k items from a (possibly initially unknown) weighted population, and is compatible for the streaming setting as well. For the weighted sampling of k items we run k concurrent independent instances, each to sample one item (cf. [74] Section 4). To sample one location from the set of non-empty locations, or to sample one realization requires time at most $O(dnz)$.

Next theorem states the quality of Algorithm 9, and is the main result of Subsection 6.3.1.

Theorem 6.17. *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. Let $\tilde{c} \in \mathbb{R}^d$ denote the output of Algorithm 9 on input $\mathcal{D}$, and let the approximation parameter be $0 < \varepsilon < 1/9$. Then, with constant probability, the output is a $(1+\varepsilon)$-approximation for the probabilistic smallest enclosing ball problem, i.e., it holds that*

$$\mathbb{E}_X\left[m(\tilde{c}, X)\right] \leq (1+\varepsilon) \min_{c \in \mathbb{R}^d} \mathbb{E}_X\left[m(c, X)\right].$$

The running time of Algorithm 9 is $O\left(dn \cdot \left((z/\varepsilon^3) \cdot \log(1/\varepsilon) + (1/\varepsilon^4) \cdot \log^2(1/\varepsilon)\right)\right)$.

Proof. The correctness of the algorithm follows from the correctness of Algorithm 7, shown by Theorem 6.5, and replacing the grid search by the subgradient algorithm from Theorem 6.15. It remains to analyze the running time.

In the first case we go through all input distributions and use k independent copies of a weighted reservoir sampler of Efraimidis [74], to get the k samples (of locations). This takes $O(dnz \cdot k) \subseteq O\left((dnz/\varepsilon^2) \cdot \log(1/\varepsilon)\right)$ time. The subsampled problem is then solved using Theorem 6.15 with $n = 1$, in time $O\left((d/\varepsilon^4) \cdot \log^2(1/\varepsilon)\right)$, with failure probability at most $\gamma = \sum_{i=1}^{5} \gamma_i \leq 6/8$ (the failure probability of Theorem 6.15).

In the second case, each realization can be sampled in time $O(dnz)$, but the probability $p_{P \neq \emptyset}$ that a realization P is non-empty can only be lower bounded by ε. We define k random variables Y_i. Let Y_1 contain the number of realizations we need to sample in order to get the first non-empty realization, what we consider as success. Analogously, let Y_i, $2 \leq i \leq k$, contain the number of samples needed to get the first non-empty realization after the $(i-1)$-th non-empty sampled realization. Each Y_i is independent geometric random variable (cf. Definition 2.24), whose success probability is $p_{P \neq \emptyset} \geq \varepsilon$. Then, using Equation (2.18), we have that the expected number of samples that we need to take in order to have k non-empty realizations is

$$\mathbb{E}\left[\sum_{i=1}^{k} Y_i\right] = \sum_{i=1}^{k} \mathbb{E}\left[Y_i\right] = \sum_{i=1}^{k} \frac{1}{p_{P \neq \emptyset}} \leq \frac{k}{\varepsilon}. \tag{6.24}$$

Thus, by an application of Markov's inequality (Lemma 2.26), the probability that we need more than $8k/\varepsilon \in O\left((1/\varepsilon^3) \cdot \log(1/\varepsilon)\right)$ trials to have k non-empty realizations is bounded by

$$\mathbf{Pr}\left[\sum_{i=1}^{k} Y_i \geq 8\frac{k}{\varepsilon}\right] \leq \frac{\mathbb{E}\left[\sum_{i=1}^{k} Y_i\right]}{8\frac{k}{\varepsilon}} \overset{(6.24)}{\leq} \frac{1}{8} = \gamma_6. \tag{6.25}$$

We can assume with constant probability that this step succeeds in $O\left(\left(dnz/\varepsilon^3\right) \cdot \log(1/\varepsilon)\right)$ time. The subsampled problem is then solved using Theorem 6.15 in $O\left(\left(dn/\varepsilon^4\right) \cdot \log^2(1/\varepsilon)\right)$ time. The failure probability in the second case is at most $\gamma = \sum_{i=1}^{6} \gamma_i \leq 7/8$. □

Comparing the result of Theorem 6.17 to the result of Theorem 6.5, the running time is reduced from $O\left(dnz/\varepsilon^{O(1)} + 1/\varepsilon^{O(d)}\right)$ to $O(dnz/\varepsilon^{O(1)})$, i.e., our dependence on the dimension d is no longer exponential but only linear. Note, in particular, that the factor of d plays a role only in computations of distances between two points in $\mathbb{R}^d$. Further, the sample size and the number of centers that need to be evaluated do not depend on the dimension d any more. This will be crucial in the next application.

6.3.2 Probabilistic support vector data description

In this subsection we want to show how to find a $(1+\varepsilon)$-approximation for the pSVDD problem (cf. Definition 6.4). Explicitly computing any center $c \in \mathcal{H}$ takes $\Omega(D)$ time and space which is prohibitive not only when $D = \infty$. Note that for the SEB and SVDD problems, any reasonable center lies in the convex hull (cf. Equation (2.8)) of the input points. Since taking the expectation is simply another linear combination over such centers, we can express any center $c \in \mathcal{H}$ as a linear combination of the elements of the set Q of non-empty locations, i.e.,

$$c = \sum_{q_{u,v} \in Q} \lambda_{u,v} \phi(q_{u,v}). \tag{6.26}$$

The idea is to exploit the characterization of Equation (6.26) to simulate Algorithm 8, and thereby Algorithm 9, to work in the feature space $\mathcal{H}$ by computing the centers and distances only implicitly.

For now, assume that any distance computation can be determined. Note that sampling a set $P_i \subset \mathbb{R}^d$ is the same as sampling the set $\phi(P_i)$ of corresponding points in $\mathcal{H}$ from the same distribution. We assume that we have a set of locations or realizations $\mathcal{P} = \{P_1, \ldots, P_N\}$, with $P_i \subset \mathbb{R}^d$. The remaining steps are passed to Theorem 6.15, which is based on Algorithm 8. First, we show the following invariant.

Lemma 6.18. *Each center c_i, $i \in \{0, 1, 2, \ldots\}$, reached during the calls to Algorithm 8, can be updated such that a linear combination $c_i = \sum_{u,v} \lambda_{u,v} \phi(q_{u,v})$ is maintained, where at most $i + 1$ terms have $\lambda_{u,v} \neq 0$.*

Proof. We prove the lemma by induction over i.

- The initial center $c_0 \in \mathcal{H}$ is chosen by sampling uniformly at random a set $P \in \mathcal{P}$ using Lemma 6.12. We take any point $q \in P$, $q \neq \perp$, which maps to $c_0 = \phi(q)$. Thus the invariant is satisfied at the beginning, where the corresponding coefficient is $\lambda = 1$ and all other coefficients are zero.
- In each iteration we randomly sample a set $P \in \mathcal{P}$ to simulate the approximate subgradient $\tilde{g}(c_i)$ at the current point c_i. The vector $\tilde{g}(c_i)$ is a vector between c_i and some point $p_{j,k} = \phi(q_{j,k}) \in \mathcal{H}$, such that $q_{j,k}$ maximizes $\|c_i - \phi(q')\|$ over all $q' \in P$ (cf. Lemma 6.10).

  To implicitly update to the next center c_{i+1} note that (cf. Algorithm 8)

$$c_{i+1} = c_i - s \cdot \frac{c_i - \phi(q_{j,k})}{\|c_i - \phi(q_{j,k})\|} = \left(1 - \frac{s}{\|c_i - \phi(q_{j,k})\|}\right) \cdot c_i + \frac{s}{\|c_i - \phi(q_{j,k})\|} \cdot \phi(q_{j,k}).$$

  Assume the invariant was valid, that c_i was represented as $c_i = \sum_{u,v} \lambda_{u,v} \phi(q_{u,v})$ with at most $i + 1$ non-zero coefficients. Then it also holds for the point c_{i+1}, since the previous non-zero coefficients of $\phi(q_{u,v})$ are multiplied by $1 - s/\|c_i - \phi(q_{j,k})\|$ and the newly added $\phi(q_{j,k})$ is assigned the coefficient $s/\|c_i - \phi(q_{j,k})\|$. So there are at most $i + 2$ non-zero coefficients. □

Lemma 6.18 implies that we do not have to store the points c_i explicitly while performing Algorithm 8. The implicit representation can be maintained using a list that stores points that appear in the approximate subgradients and their corresponding non-zero coefficients.

To actually compute the coefficients, we need to be able to compute Euclidean distances as well as determine s. Using Lemma 6.13 we determined the step size s using an estimator $\tilde{R} = \sum_{P \in \mathcal{S}} m(c_0, P)$ based on a small sample $\mathcal{S}$. In particular, this requires distance computations again. To this end, we show how to compute $\|c_i - \phi(q)\|$ for any location $q \in \mathbb{R}^d$. Recall that the kernel function implicitly defines the inner product in $\mathcal{H}$. Therefore, it holds that

$$\begin{aligned}\|c_i - \phi(q)\|^2 &= \left\|\sum_{u,v} \lambda_{u,v}\phi(q_{u,v}) - \phi(q)\right\|^2 = \left\|\sum_{w=0}^{i} \lambda_w \phi(q_w) - \phi(q)\right\|^2 \\ &= \left\|\sum_{w=0}^{i} \lambda_w \phi(q_w)\right\|^2 + \|\phi(q)\|^2 - 2\sum_{w=0}^{i} \lambda_w \langle \phi(q_w), \phi(q)\rangle \\ &= \sum_{w=0}^{i}\sum_{w'=0}^{i} \lambda_w \lambda_{w'} K(q_w, q_{w'}) + K(q,q) - 2\sum_{w=0}^{i} \lambda_w K(q_w, q), \end{aligned} \tag{6.27}$$

where $w, w' \in \{0, \ldots, i\}$ index the locations $q_w, q_{w'}$ with corresponding $\lambda_w, \lambda_{w'} \neq 0$ in iteration i. Therefore, we have the following theorem.

Theorem 6.19. *Let $\mathcal{D}$ be a set of n discrete distributions, where each distribution is defined over z locations in $\mathbb{R}^d \cup \{\perp\}$. There exists an algorithm that implicitly computes $\tilde{c} \in \mathcal{H}$ that with constant probability is a $(1+\varepsilon)$-approximation for the probabilistic support vector data description problem, i.e., it holds that*

$$\mathbb{E}_X\left[m(\tilde{c}, \phi(X))\right] \leq (1+\varepsilon) \min_{c \in \mathcal{H}} \mathbb{E}_X\left[m(c, \phi(X))\right],$$

where the expectation is taken over the randomness of $X \sim \mathcal{D}$. The running time of the algorithm is $O\left(dn \cdot \left(\left(z/\varepsilon^3\right) \cdot \log(1/\varepsilon) + \left(1/\varepsilon^8\right) \cdot \log^2(1/\varepsilon)\right)\right)$.

Proof. With the described adaptations by Lemma 6.18 and Equation (6.27), the correctness of the algorithm follows from Theorem 6.17.

The running time increases by a factor that is imposed by the simulation of the distance computations within Algorithm 8. Note that by the invariant there are at most $i+1$ non-zero coefficients in the i-th step. Equation (6.27) can thus be evaluated in time $O\left(i^2 d\right)$, assuming K can be evaluated in time $O(d)$. We conclude:

- The sampling part of Algorithm 9 does not change, and runs in $O\left(\left(dnz/\varepsilon^3\right) \cdot \log(1/\varepsilon)\right)$ time.
- Estimating $\tilde{R} = \sum_{P \in \mathcal{S}} m(c_0, P)$ takes time $O(dn/\varepsilon)$, since $i = 0$, $|\mathcal{S}| = O(1/\varepsilon)$, and for each P we have $|P| \leq n$.
- The subgradient computation in the i-th iteration of the main loop takes $O\left(dni^2\right)$ time, since it needs to maximize over n distances. This means that for $\ell \in O\left(1/\varepsilon^2\right)$ iterations we need $O\left(dn\sum_{i=1}^{\ell} i^2\right) = O\left(dn\ell^3\right) = O\left(dn/\varepsilon^6\right)$ time. This is repeated $O(\log(1/\varepsilon))$ times, which implies a running time of $O\left(\left(dn/\varepsilon^6\right) \cdot \log(1/\varepsilon)\right)$.
- The evaluation of the minimum at the end reaches, as before, over $O\left(\left(1/\varepsilon^2\right) \cdot \log(1/\varepsilon)\right)$ centers, each defined by $O(\ell)$ non-zero coefficients. Each is evaluated with respect

to a sample of $O\left((1/\varepsilon^2)\cdot\log(1/\varepsilon)\right)$ sets from the input. Since each maximum distance evaluation takes $O\left(dn\ell^2\right) = O\left(dn/\varepsilon^4\right)$ time, we have that the total time for evaluation is $O\left(\left(dn/\varepsilon^8\right)\cdot\log^2(1/\varepsilon)\right)$.

Summing the running times yields the claim. □

6.4 Conclusion and open questions

We studied in this chapter a generalization of the 1-center and the 1-median problems – the set median problem in high dimensions, that minimizes the sum of distances to the farthest point in each input set. We presented a $(1+\varepsilon)$-approximation algorithm whose running time is linear in d and independent of the number of input sets. We further discussed that in high dimensions the dependence on the size of the input sets cannot be reduced sublinearly without losing a factor of roughly $\sqrt{2}$.

Our result resolves an open problem, posed in [145], and improves the previously best algorithm for the probabilistic smallest enclosing ball problem in high dimensions by reducing the dependence on d from exponential to linear. This enables running the algorithm in high dimensional Hilbert spaces induced by kernel functions, which makes it more flexible and viable as a building block in machine learning and data analysis. As an example we transferred the kernel based SVDD problem of Tax and Duin [166] to the probabilistic data setting.

Our algorithms assume discrete input distributions. It would be interesting to extend the algorithms presented in this chapter to various continuous distributions. There are no known results for a notion of probabilistic points that would be defined through continuous distributions.

The pSEB problem minimizes the expected maximum distance. When it comes to minimizing volumes of balls or in the context of Gaussian distributions it might be interesting to study higher moments of the maximum distance. This corresponds to a generalization of the set median problem to minimizing the sum of higher powers of maximum distances. This topic is completely open for further research.

Finally, as we did for the probabilistic SVDD problem, we hope that the methods presented in this chapter may help to extend more shape fitting and machine learning problems to the probabilistic setting.

A Additional analysis to Lemma 5.1

In the proof of Lemma 5.1 some analysis was shortened for the sake of readability. Here we present the exact computation. We apply the method of integration by parts, for all integrals in the following four claims.

Claim A.1. *It holds for* $\varphi \in [0,1]$ *that:*

$$\int_0^{\arccos\varphi} \sin^2(x)dx = \frac{1}{2}\left(\arccos\varphi - \varphi\cdot\sqrt{1-\varphi^2}\right). \tag{A.1}$$

Proof. We denote the value of the integral in Equation (A.1) with A. Then

$$\begin{aligned} A = \int_0^{\arccos\varphi} \sin^2(x)dx &= -\sin(x)\cos(x)\big|_0^{\arccos\varphi} - \int_0^{\arccos\varphi} -\cos^2(x)dx \\ &= -\varphi\cdot\sin(\arccos\varphi) + \int_0^{\arccos\varphi} \left(1-\sin^2(x)\right)dx \\ &= -\varphi\cdot\sqrt{1-\varphi^2} + \arccos\varphi - A. \end{aligned} \tag{A.2}$$

Equation (A.2) implies the correctness of the claim. □

Claim A.2. *It holds for* $\varphi \in [0,1]$ *that:*

$$\int_0^{\arccos\varphi} \sin^3(x)dx = \frac{1}{3}\varphi^3 - \varphi + \frac{2}{3}. \tag{A.3}$$

Proof. We denote the value of the integral in Equation (A.3) with A. Then

$$\begin{aligned} A = \int_0^{\arccos\varphi} \sin^3(x)dx &= -\sin^2(x)\cos(x)\big|_0^{\arccos\varphi} - \int_0^{\arccos\varphi} -\cos(x)\cdot 2\sin(x)\cos(x)dx \\ &= \left(\cos^3(x)-\cos(x)\right)\big|_0^{\arccos\varphi} + 2\int_0^{\arccos\varphi} \sin(x)\cos^2(x)dx \\ &= \varphi^3 - \varphi + 2\int_0^{\arccos\varphi} \sin(x)\cos^2(x)dx. \end{aligned} \tag{A.4}$$

We denote the integral in Equation (A.4) with B. Then, we have

$$B = \int_0^{\arccos\varphi} \sin(x)\cos^2(x)dx = -\cos^3(x)\big|_0^{\arccos\varphi} - 2\int_0^{\arccos\varphi} \sin(x)\cos^2(x)dx$$
$$= -\varphi^3 + 1 - 2B. \tag{A.5}$$

By taking the value of B from Equation (A.5), and substituting into Equation (A.4), we obtain Equation (A.3), as claimed. □

Claim A.3. *It holds for $\varphi \in [0,1]$ that:*

$$\int_0^{\arccos\varphi} \sin^4(x)dx = \frac{1}{8}\left(3\arccos\varphi + \varphi\cdot\sin(\arccos\varphi)\cdot\left(2\varphi^2 - 5\right)\right). \tag{A.6}$$

Proof. We denote the value of the integral in Equation (A.6) with A. Then

$$A = \left(-\sin^3(x)\cos(x)\right)\big|_0^{\arccos\varphi} + 3\int_0^{\arccos\varphi} \sin^2(x)\cos^2(x)dx$$
$$= -\varphi\sin^3(\arccos\varphi) + 3\int_0^{\arccos\varphi} \sin^2(x)\left(1-\sin^2(x)\right)dx$$
$$= -\varphi(1-\varphi^2)\sin(\arccos\varphi) + 3\int_0^{\arccos\varphi} \sin^2(x)dx - 3A. \tag{A.7}$$

We denote the integral on the right-hand side of Equation (A.7) with B. Then, we have

$$B = \int_0^{\arccos\varphi} \sin^2(x)dx = (-\sin(x)\cos(x))\big|_0^{\arccos\varphi} + \int_0^{\arccos\varphi} \cos^2(x)dx$$
$$= -\varphi\sin(\arccos\varphi) + \arccos\varphi - B. \tag{A.8}$$

By taking the value of B from Equation (A.8) into Equation (A.7), we obtain

$$4A = -\varphi(1-\varphi^2)\sin(\arccos\varphi) + \frac{3}{2}\left(-\varphi\sin(\arccos\varphi) + \arccos\varphi\right),$$

and this implies Equation (A.6), as claimed. □

Claim A.4. *It holds for $\varphi \in [0,1]$ that:*

$$\int_0^{\arccos\varphi} \sin^5(x)dx = \frac{1}{15}\left(-3\varphi^5 + 10\varphi^3 - 15\varphi + 8\right). \tag{A.9}$$

Proof. We denote the value of the integral in Equation (A.9) with A. Then,

$$\begin{aligned} A &= \left(-\sin^4(x)\cos(x)\right)\Big|_0^{\arccos\varphi} + 4\int_0^{\arccos\varphi} \sin^3(x)\cos^2(x)dx \\ &= -\varphi\left(1-\varphi^2\right)^2 + 4\int_0^{\arccos\varphi} \sin^3(x)\cos^2(x)dx. \end{aligned} \tag{A.10}$$

We denote the integral on the right-hand side in Equation (A.10) with B. Then, we have

$$\begin{aligned} B &= \left(\cos^5(x) - \cos^3(x)\right)\Big|_0^{\arccos\varphi} + \int_0^{\arccos\varphi} \cos(x)\cdot\left(4\sin(x)\cos^3(x) - 2\sin(x)\cos(x)\right)dx \\ &= \varphi^5 - \varphi^3 + 4\int_0^{\arccos\varphi} \sin(x)\left(1-\sin^2(x)\right)^2 dx - 2\int_0^{\arccos\varphi} \sin(x)\left(1-\sin^2(x)\right)dx \\ &= \varphi^5 - \varphi^3 + 2\int_0^{\arccos\varphi} \sin(x)dx - 6\int_0^{\arccos\varphi} \sin^3(x)dx + 4\int_0^{\arccos\varphi} \sin^5(x)dx \\ &\overset{(A.3)}{=} \varphi^5 - \varphi^3 + 2\left(1-\varphi\right) - 6\left(\frac{1}{3}\varphi^3 - \varphi + \frac{2}{3}\right) + 4A. \end{aligned} \tag{A.11}$$

By substituting the value of B from Equation (A.11) into Equation (A.10), we obtain the correctness of Equation (A.9). □

In the next three claims we show the inequalities needed in the proof of Lemma 5.1. In each of them we state a function f that is monotonically increasing on $[0,\pi]$, with $f(0)=0$.

Claim A.5. *It holds for $x \in [0,1]$ that*

$$1 - \frac{3}{2}\cdot\left(\frac{1}{3}x^3 - x + \frac{2}{3}\right) \leq \left(1+\frac{2}{\pi}\right)\cdot x. \tag{A.12}$$

Proof. We observe the function $f(x) = x^3 + \left(\frac{4}{\pi} - 1\right)x$. By rearranging the expression in Equation (A.12) we obtain that (A.12) is equivalent to $f(x) \geq 0$, for $x \in [0,1]$. It is $f(0) = 0$. The first derivation of f is $f'(x) = 3x^2 + \frac{4}{\pi} - 1 > 0$, for all $x \in \mathbb{R}$. Therefore, $f(x) \geq 0$ for all $x \in [0,1]$, and the correctness of Claim A.5 is proven. □

Claim A.6. *It holds for $x \in [0,1]$ that*

$$1 - \frac{2}{3\pi}\left(3\arccos(x) + x\cdot\sin(\arccos(x))\cdot\left(2x^2-5\right)\right) \leq \frac{16}{3\pi}x. \tag{A.13}$$

Proof. Equation (A.13) is equivalent to $f(x) \geq 0$, for $x \in [0,1]$, where the function $f(x)$ is $f(x) = \frac{16}{3\pi}x - 1 + \frac{2}{3\pi} \cdot \big(3\arccos(x) + x \cdot \sin(\arccos(x)) \cdot \big(2x^2 - 5\big)\big)$. It is $f(0) = 0$. Note that $\sin(\arccos(x)) = \sqrt{1-x^2}$, for $x \in [0,1]$. Then, we have

$$\begin{aligned} f'(x) &= \frac{16}{3\pi} + \frac{2}{3\pi} \cdot \left(\frac{-3}{\sqrt{1-x^2}} + \big(6x^2 - 5\big)\sin(\arccos(x)) + \big(2x^3 - 5x\big) \cdot x \cdot \frac{-1}{\sqrt{1-x^2}} \right) \\ &= \frac{16}{3\pi} + \frac{2}{3\pi} \cdot \left(\frac{\big(2x^2 - 3\big)\big(1 - x^2\big)}{\sqrt{1-x^2}} + \big(6x^2 - 5\big)\sqrt{1-x^2} \right) \\ &= \frac{16}{3\pi} + \frac{16}{3\pi} \cdot \sqrt{1-x^2} \cdot \big(x^2 - 1\big) \geq \frac{16}{3\pi} - \frac{16}{3\pi} = 0, \end{aligned}$$

for $x \in [0,1]$. This implies that $f(x) \geq 0$, as claimed. □

Claim A.7. *It holds for $x \in [0,1]$ that*

$$1 - \frac{1}{8}\big(-3x^5 + 10x^3 - 15x + 8\big) \leq \frac{15}{8}x. \tag{A.14}$$

Proof. Equation (A.14) can be transformed into $3x^5 - 10x^3 + 15x \leq 15x$. Since for $x \in [0,1]$, it is $3x^5 \leq 10x^3$, Equation (A.14) is satisfied for all $x \in [0,1]$. We note that the constant $\frac{15}{8}$ on the right-hand side of Equation (A.14) is the best possible for $x \in [0,1]$. Namely, for the function $f(x) = \alpha x - \frac{1}{8}\big(3x^5 - 10x^3 + 15x\big)$, for some constant $\alpha \geq 0$, it holds that $f'(x) = \alpha - \frac{15}{8}\big(x^2 - 1\big)^2$. For $x \in [0,1]$, it is $f'(x) \geq 0$ for $\alpha \geq \frac{15}{8}$. □

Bibliography

[1] M. Abam, M. de Berg, P. Hachenberger, and A. Zarei. Streaming algorithms for line simplification. *Discrete & Computational Geometry*, 43:497–515, 2010. Previously appeared in the 23rd ACM Symposium on Computational Geometry, SoCG 2007.

[2] A. Abboud, A. Bačkurs, and V. V. Williams. Tight hardness results for LCS and other sequence similarity measures. In V. Guruswami, editor, *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2015*, pages 59–78, 2015.

[3] M. R. Ackermann, J. Blömer, and C. Sohler. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms*, 6(4):59:1–59:26, 2010. Previously appeared in the 19th ACM-SIAM Symposium on Discrete Algorithms, SODA 2008.

[4] P. Afshani and A. Driemel. On the complexity of range searching among curves. In *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 898–917, 2018.

[5] P. K. Agarwal, R. Ben Avraham, H. Kaplan, and M. Sharir. Computing the discrete Fréchet distance in subquadratic time. *SIAM Journal on Computing*, 43(2):429–449, 2014. Previously appeared in the 24th ACM-SIAM Symposium on Discrete Algorithms, SODA 2013.

[6] P. K. Agarwal, K. Fox, J. Pan, and R. Ying. Approximating dynamic time warping and edit distance for a pair of point sequences. In S. Fekete and A. Lubiw, editors, *Proceedings of the 32nd International Symposium on Computational Geometry, SoCG*, volume 51 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 6:1–6:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[7] P. K. Agarwal, S. Har-Peled, N. H. Mustafa, and Y. Wang. Near-linear time approximation algorithms for curve simplification. *Algorithmica*, 42:203–219, 2005. Previously appeared in the 10th Annual European Symposium on Algorithms, ESA 2002.

[8] P. K. Agarwal, S. Har-Peled, K. R. Varadarajan, et al. Geometric approximation via coresets. *Combinatorial and computational geometry*, 52:1–30, 2005.

[9] P. K. Agarwal and R. Sharathkumar. Streaming algorithms for extent problems in high dimensions. *Algorithmica*, 72(1):83–98, 2015. Previously appeared in the 21st ACM-SIAM Symposium on Discrete Algorithms, SODA 2010.

[10] S. R. Aghabozorgi, A. S. Shirkhorshidi, and Y. W. Teh. Time-series clustering - A decade review. *Information Systems*, 53:16–38, 2015.

[11] S. Ahmadian, A. Norouzi-Fard, O. Svensson, and J. Ward. Better guarantees for k-means and euclidean k-median by primal-dual algorithms. In C. Umans, editor, *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 61–72, 2017.

[12] H. Ahn, H. Alt, M. Buchin, E. Oh, L. Scharf, and C. Wenk. Middle curves based on discrete Fréchet distance. *Computational Geometry: Theory and Applications*, 89:101621, 2020. Previously appeared in the 12th Latin American Conference on Theoretical Informatics, LATIN 2016.

[13] H. Alt. The computational geometry of comparing shapes. In S. Albers, H. Alt, and S. Näher, editors, *Efficient Algorithms, Essays Dedicated to Kurt Mehlhorn on the Occasion of His 60th Birthday*, volume 5760 of *Lecture Notes in Computer Science*, pages 235–248. Springer, 2009.

[14] H. Alt and M. Godau. Computing the Fréchet distance between two polygonal curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.

[15] H. Alt, C. Knauer, and C. Wenk. Matching polygonal curves with respect to the Fréchet distance. In *Proceedings of the 18th Symposium on Theoretical Aspects of Computer Science, STACS*, volume 2010 of *Lecture Notes in Computer Science*, pages 63–74, 2001.

[16] V. Arya, N. Garg, R. Khandekar, A. Meyerson, K. Munagala, and V. Pandit. Local search heuristics for k-median and facility location problems. *SIAM Journal on Computing*, 33(3):544–562, 2004. Previously appeared in the 33rd ACM Symposium on Theory of Computing, STOC 2001.

[17] R. A. Askey and R. Roy. Gamma function. *NIST handbook of mathematical functions, US Department of Commerce, Washington, D.C.*, pages 135–147, 2010.

[18] A. Bačkurs and A. Sidiropoulos. Constant-distortion embeddings of Hausdorff metrics into constant-dimensional ℓ_p spaces. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 1:1–1:15, 2016.

[19] M. Bădoiu, J. Chuzhoy, P. Indyk, and A. Sidiropoulos. Low-distortion embeddings of general metrics into the line. In *Proceedings of the 37th ACM Symposium on Theory of Computing, STOC*, pages 225–233, 2005.

[20] M. Bădoiu and K. L. Clarkson. Smaller core-sets for balls. In *Proceedings of the 14th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 801–802, 2003.

[21] M. Bădoiu and K. L. Clarkson. Optimal core-sets for balls. *Computational Geometry: Theory and Applications*, 40(1):14–22, 2008.

[22] M. Bădoiu, S. Har-Peled, and P. Indyk. Approximate clustering via core-sets. In *Proceedings of the 34th ACM Symposium on Theory of Computing, STOC*, pages 250–257, 2002.

[23] Y. Bartal and N. Fandina. Course notes in Metric embedding theory and its algorithmic applications, CS-67720, fall 2017/2018, lecture 1, 2017.

[24] Y. Bartal, L. Gottlieb, and O. Neiman. On the impossibility of dimension reduction for doubling subsets of ℓ_p. *SIAM Journal on Discrete Mathematics*, 29(3):1207–1222, 2015. Previously appeared in the 30th ACM Symposium on Computational Geometry, SoCG 2014.

[25] A. Beck and S. Sabach. Weiszfeld's method: Old and new results. *Journal of Optimization Theory and Applications*, pages 1–40, 2014.

[26] S. Bereg, M. Jiang, W. Wang, B. Yang, and B. Zhu. Simplifying 3D polygonal chains under the discrete Fréchet distance. In E. S. Laber, C. F. Bornstein, L. T. Nogueira, and L. Faria, editors, *Proceedings of the 8th Latin American Conference on Theoretical Informatics, LATIN*, volume 4957 of *Lecture Notes in Computer Science*, pages 630–641. Springer, 2008.

[27] E. Bingham, A. Gionis, N. Haiminen, H. Hiisilä, H. Mannila, and E. Terzi. Segmentation and dimensionality reduction. In *Proceedings of the 6th SIAM International Conference on Data Mining, SDM*, pages 372–383, 2006.

[28] A. Blum, J. Hopcroft, and R. Kannan. *Foundations of Data Science*. Cambridge University Press, 2020.

[29] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[30] M. Brill, T. Fluschnik, V. Froese, B. J. Jain, R. Niedermeier, and D. Schultz. Exact mean computation in dynamic time warping spaces. *Data Mining and Knowledge Discovery*, 33(1):252–291, 2019. Previously appeared in the SIAM International Conference on Data Mining, SDM 2018.

[31] K. Bringmann. Why walking the dog takes time: Fréchet distance has no strongly subquadratic algorithms unless SETH fails. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 661–670, 2014.

[32] K. Bringmann and B. R. Chaudhury. Polyline simplification has cubic complexity. In G. Barequet and Y. Wang, editors, *Proceedings of the 35th International Symposium on Computational Geometry, SoCG*, volume 129 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[33] K. Bringmann and M. Künnemann. Quadratic conditional lower bounds for string problems and dynamic time warping. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 79–97, 2015.

[34] K. Bringmann and M. Künnemann. Improved approximation for Fréchet distance on c-packed curves matching conditional lower bounds. *International Journal of Computational Geometry & Applications*, 27(1-2):85–120, 2017. Previously appeared in the 26th International Symposium on Algorithms and Computation, ISAAC 2015.

[35] K. Bringmann, M. Künnemann, and A. Nusser. Walking the dog fast in practice: Algorithm engineering of the fréchet distance. In G. Barequet and Y. Wang, editors, *Proceedings of the 35th International Symposium on Computational Geometry, SoCG*, pages 17:1–17:21, 2019.

[36] K. Bringmann, M. Künnemann, and A. Nusser. When Lipschitz walks your dog: Algorithm engineering of the discrete Fréchet distance under translation. In F. Grandoni, G. Herman, and P. Sanders, editors, *Proceedings of the 28th Annual European Symposium on Algorithms, ESA 2020*, volume 173 of *LIPIcs*, pages 25:1–25:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[37] K. Bringmann and W. Mulzer. Approximability of the discrete Fréchet distance. *Journal of Computational Geometry*, 7(2):46–76, 2016. Previously appeared in the 31st International Symposium on Computational Geometry, SoCG 2015.

[38] K. Buchin, M. Buchin, C. Knauer, G. Rote, and C. Wenk. How difficult is it to walk the dog? In *Proceedings of the 23rd European Workshop on Computational Geometry*, pages 170–173, 2007.

[39] K. Buchin, M. Buchin, M. Konzack, W. Mulzer, and A. Schulz. Fine-grained analysis of problems on curves. In *Proceedings of the 32nd European Workshop on Computational Geometry*, pages 19–22, 2016.

[40] K. Buchin, M. Buchin, W. Meulemans, and W. Mulzer. Four Soviets walk the dog: Improved bounds for computing the Fréchet distance. *Discrete & Computational Geometry*, 58(1):180–216, 2017. Previously appeared in the 25th ACM-SIAM Symposium on Discrete Algorithms, SODA 2014.

[41] K. Buchin, M. Buchin, M. van Kreveld, M. Löffler, R. I. Silveira, C. Wenk, and L. Wiratma. Median trajectories. *Algorithmica*, 66(3):595–614, 2013. Previously appeared in the 18th Annual European Symposium on Algorithms, ESA 2010.

[42] K. Buchin, M. Buchin, and Y. Wang. Exact algorithms for partial curve matching via the Fréchet distance. In *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 645–654, 2009.

[43] K. Buchin, J. Chun, M. Löffler, A. Markovic, W. Meulemans, Y. Okamoto, and T. Shiitada. Folding free-space diagrams: Computing the Fréchet distance between 1-dimensional curves (multimedia contribution). In *Proceedings of the 33rd International Symposium on Computational Geometry, SoCG*, pages 64:1–64:5, 2017.

[44] K. Buchin, A. Driemel, J. Gudmundsson, M. Horton, I. Kostitsyna, M. Löffler, and M. Struijs. Approximating (k, ℓ)-center clustering for curves. In T. M. Chan, editor, *Proceedings of the 30th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2922–2938, 2019.

[45] K. Buchin, A. Driemel, and M. Struijs. On the hardness of computing an average curve. In S. Albers, editor, *Proceedings of the 17th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT*, volume 162 of *LIPIcs*, pages 19:1–19:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[46] M. Buchin, A. Driemel, and D. Rohde. Approximating (k, ℓ)-median clustering for polygonal curves. In D. Marx, editor, *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2697–2717, 2021.

[47] M. Buchin, A. Driemel, and B. Speckmann. Computing the Fréchet distance with shortcuts is NP-hard. In S. Cheng and O. Devillers, editors, *Proceedings of the 30th ACM Symposium on Computational Geometry, SoCG*, page 367. ACM, 2014.

[48] M. Buchin, N. Funk, and A. Krivošija. On the complexity of the middle curve problem. *CoRR*, abs/2001.10298, 2020.

[49] L. Bulteau, V. Froese, and R. Niedermeier. Hardness of consensus problems for circular strings and time series averaging. *CoRR*, abs/1804.02854, 2018.

[50] J. Byrka, T. Pensyl, B. Rybicki, A. Srinivasan, and K. Trinh. An improved approximation for k-median and positive correlation in budgeted optimization. *ACM Transactions on Algorithms*, 13(2):23:1–23:31, 2017.

[51] M. Ceccarello, A. Driemel, and F. Silvestri. FRESH: Fréchet similarity with hashing. In Z. Friggstad, J. Sack, and M. R. Salavatipour, editors, *Proceedings of the 16th Algorithms and Data Structures Symposium - WADS (formerly Workshop on Algorithms and Data Structures)*, pages 254–268, 2019.

[52] M. T. Chao. A general purpose unequal probability sampling plan. *Biometrika*, 69(3):653–656, 1982.

[53] M. Charikar, S. Guha, É. Tardos, and D. B. Shmoys. A constant-factor approximation algorithm for the k-median problem. *Journal of Computer and System Sciences*, 65(1):129–149, 2002. Previously appeared in the 31st ACM Symposium on Theory of Computing, STOC 1999.

[54] K. Chen. On coresets for k-median and k-means clustering in metric and Euclidean spaces and their applications. *SIAM Journal on Computing*, 39(3):923–947, 2009. Previously appeared in the 17th ACM-SIAM Symposium on Discrete Algorithms, SODA 2006.

[55] J.-M. Chiou and P.-L. Li. Functional clustering and identifying substructures of longitudinal data. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):679–699, 2007.

[56] M. Cieliebak, P. Flocchini, G. Prencipe, and N. Santoro. Distributed computing by mobile robots: Gathering. *SIAM Journal on Computing*, 41(4):829–879, 2012.

[57] M. B. Cohen, Y. T. Lee, G. L. Miller, J. Pachocki, and A. Sidford. Geometric median in nearly linear time. In *Proceedings of the 48th ACM Symposium on Theory of Computing, STOC*, pages 9–21, 2016.

[58] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms.* MIT Press, 3rd edition, 2009.

[59] G. Cormode and A. McGregor. Approximation algorithms for clustering uncertain data. In *Proceedings of the 27th ACM Symposium on Principles of Database Systems, PODS*, pages 191–200, 2008.

[60] C. Daskalakis, I. Diakonikolas, and M. Yannakakis. How good is the chord algorithm? *SIAM Journal on Computing*, 45(3):811–858, 2016.

[61] M. de Berg, A. F. Cook, and J. Gudmundsson. Fast Fréchet queries. *Computational Geometry: Theory and Applications*, 46(6):747–755, 2013. Previously appeared in the 22nd International Symposium on Algorithms and Computation, ISAAC 2011.

[62] H. Ding, G. Trajcevski, P. Scheuermann, X. Wang, and E. J. Keogh. Querying and mining of time series data: Experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

[63] D. H. Douglas and T. K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Classics in Cartography: Reflections on Influential Articles from Cartographica*, pages 15–28, 2011.

[64] A. Driemel. *Realistic analysis for algorithmic problems on geographical data.* PhD thesis, Utrecht University, 2013.

[65] A. Driemel and S. Har-Peled. Jaywalking your dog – Computing the Fréchet distance with shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013. Previously appeared in the 23rd ACM-SIAM Symposium on Discrete Algorithms, SODA 2012.

[66] A. Driemel, S. Har-Peled, and C. Wenk. Approximating the Fréchet distance for realistic curves in near-linear time. *Discrete & Computational Geometry*, 48(1):94–127, 2012. Previously appeared in the 26th ACM Symposium on Computational Geometry, SoCG 2010.

[67] A. Driemel and A. Krivošija. Probabilistic embeddings of the Fréchet distance. In L. Epstein and T. Erlebach, editors, *16th International Workshop on Approximation and Online Algorithms, WAOA, Revised Selected Papers*, pages 218–237, 2018.

[68] A. Driemel, A. Krivošija, and C. Sohler. Clustering time series under the Fréchet distance. In R. Krauthgamer, editor, *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 766–785, 2016.

[69] A. Driemel and I. Psarros. $(2+\epsilon)$-ANN for time series under the Fréchet distance. *CoRR*, abs/2008.09406, 2020.

[70] A. Driemel, I. Psarros, and M. Schmidt. Sublinear data structures for short fréchet queries. *CoRR*, abs/1907.04420, 2019.

[71] A. Driemel and F. Silvestri. Locally-sensitive hashing of curves. In *Proceedings of the 33rd International Symposium on Computational Geometry, SoCG*, pages 37:1–37:16, 2017.

[72] A. Dumitrescu and G. Rote. On the Fréchet distance of a set of curves. In *Proceedings of the 16th Canadian Conference on Computational Geometry*, pages 162–165, 2004.

[73] H. Edelsbrunner and E. P. Mücke. Simulation of simplicity: a technique to cope with degenerate cases in geometric algorithms. *ACM Transactions on Graphics*, 9(1):66–104, 1990. Previously appeared in the 36th ACM Symposium on Computational Geometry, SoCG 1988.

[74] P. S. Efraimidis. Weighted random sampling over data streams. In *Algorithms, Probability, Networks, and Games*, pages 183–195. Springer International, 2015.

[75] T. Eiter and H. Mannila. Computing discrete Fréchet distance. Technical Report CD-TR 94/64, Christian Doppler Laboratory, 1994.

[76] I. Z. Emiris and I. Psarros. Products of Euclidean metrics and applications to proximity questions among curves. In B. Speckmann and C. D. Tóth, editors, *Proceedings of the 34th International Symposium on Computational Geometry, SoCG*, pages 37:1–37:13, 2018.

[77] Euclid. *Elementi I-VI.* Kruzak Zagreb, 1999. Translated from the Ancient Greek original to Croatian by Maja Hudoletnjak Grgić.

[78] C. Fan, O. Filtser, M. J. Katz, T. Wylie, and B. Zhu. On the chain pair simplification problem. In F. Dehne, J. Sack, and U. Stege, editors, *Proceedings of the 14th Algorithms and Data Structures Symposium - WADS (formerly Workshop on Algorithms and Data Structures)*, volume 9214 of *Lecture Notes in Computer Science*, pages 351–362. Springer, 2015.

[79] C. Fan, O. Filtser, M. J. Katz, and B. Zhu. On the general chain pair simplification problem. In P. Faliszewski, A. Muscholl, and R. Niedermeier, editors, *Proceedings of the 41st International Symposium on Mathematical Foundations of Computer*

Science, volume 58 of *LIPIcs*, pages 37:1–37:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[80] T. Feder and D. Greene. Optimal algorithms for approximate clustering. In *Proceedings of the 20th ACM Symposium on Theory of Computing, STOC*, pages 434–444, 1988.

[81] D. Feldman and M. Langberg. A unified framework for approximating and clustering data. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC*, pages 569–578, 2011.

[82] M. R. Fellows, F. V. Fomin, D. Lokshtanov, E. Losievskaja, F. A. Rosamond, and S. Saurabh. Distortion is fixed parameter tractable. *ACM Transactions on Computation Theory*, 5(4):16:1–16:20, 2013. Previously appeared in the 36th International Colloquium on Automata, Languages, and Programming, ICALP 2009.

[83] A. Filtser and O. Filtser. Static and streaming data structures for Fréchet distance queries. In D. Marx, editor, *Proceedings of the 32nd ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1150–1170, 2021.

[84] A. Filtser, O. Filtser, and M. J. Katz. Approximate nearest neighbor for curves - Simple, efficient, and deterministic. In A. Czumaj, A. Dawar, and E. Merelli, editors, *Proceedings of the 47th International Colloquium on Automata, Languages, and Programming, ICALP*, volume 168 of *LIPIcs*, pages 48:1–48:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[85] M. M. Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, 1906.

[86] M. Godau. A natural metric for curves – computing the distance for polygonal chains and approximation algorithms. In *Proceedings of the 8th Symposium on Theoretical Aspects of Computer Science, STACS*, pages 127–136. Springer, 1991.

[87] A. Goel, P. Indyk, and K. R. Varadarajan. Reductions among high dimensional proximity problems. In *Proceedings of the 12th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 769–778, 2001.

[88] O. Gold and M. Sharir. Dynamic time warping and geometric edit distance: Breaking the quadratic barrier. In *Proceedings of the 44th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 25:1–25:14, 2017.

[89] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science*, 38(0):293–306, 1985.

[90] A. Grønlund and S. Pettie. Threesomes, degenerates, and love triangles. *Journal of the Association for Computing Machinery*, 65(4):22:1–22:25, 2018. Previously appeared in the 55th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2014.

[91] S. Guha and K. Munagala. Exceeding expectations and clustering uncertain data. In *Proceedings of the 28th ACM Symposium on Principles of Database Systems, PODS*, pages 269–278, 2009.

[92] L. J. Guibas, J. Hershberger, J. S. B. Mitchell, and J. Snoeyink. Approximating polygons and subdivisions with minimum link paths. *International Journal of Computational Geometry & Applications*, 3(4):383–415, 1993. Previously appeared in the 2nd International Symposium on Algorithms 1991.

[93] A. Gupta, R. Krauthgamer, and J. R. Lee. Bounded geometries, fractals, and low-distortion embeddings. In *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 534–543, 2003.

[94] E. Gurarie, C. Bracis, M. Delgado, T. D. Meckley, I. Kojola, and C. M. Wagner. What is the animal doing? Tools for exploring behavioural structure in animal movements. *Journal of Animal Ecology*, 85(1):69–84, 2016.

[95] S. Har-Peled. *Geometric Approximation Algorithms*. Number 173 in Mathematical Surveys and Monographs. American Mathematical Society, 2011.

[96] S. Har-Peled and A. Kushal. Smaller coresets for k-median and k-means clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007. Previously appeared in the 21st ACM Symposium on Computational Geometry, SoCG 2005.

[97] S. Har-Peled and S. Mazumdar. On coresets for k-means and k-median clustering. In *Proceedings of the 36th ACM Symposium on Theory of Computing, STOC*, pages 291–300, 2004.

[98] S. Har-Peled and B. Raichel. The Fréchet distance revisited and extended. *ACM Transactions on Algorithms*, 10(1):3:1–3:22, 2014. Previously appeared in the 27th ACM Symposium on Computational Geometry, SoCG 2011.

[99] J. Håstad, L. Ivansson, and J. Lagergren. Fitting points on the real line and its application to RH mapping. *Journal of Algorithms*, 49(1):42–62, 2003. Previously appeared in the 6th Annual European Symposium on Algorithms, ESA 1998.

[100] J. Himberg, K. Korpiaho, H. Mannila, J. Tikanmäki, and H. T. Toivonen. Time series segmentation for context recognition in mobile devices. In *Proceedings of the 1st IEEE International Conference on Data Mining, ICDM*, pages 203–210, 2001.

[101] D. S. Hochbaum and D. B. Shmoys. A best possible heuristic for the k-center problem. *Mathematics of Operations Research*, 10(2):180–184, 1985.

[102] R. A. Horn and C. R. Johnson. *Matrix analysis*. Cambridge University Press, 2nd edition, 2013.

[103] Y.-C. Hsu and A.-P. Chen. A clustering time series model for the optimal hedge ratio decision making. *Neurocomputing*, 138(0):358–370, 2014.

[104] B. Huang and W. Kinsner. ECG frame classification using dynamic time warping. In *IEEE Canadian Conference on Electrical and Computer Engineering, CCECE*, volume 2, pages 1105–1110, 2002.

[105] L. Huang and J. Li. Stochastic k-center and j-flat-center problems. In *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 110–129, 2017.

[106] L. Huang, J. Li, J. M. Phillips, and H. Wang. ε-kernel coresets for stochastic points. In *Proceedings of the 24th Annual European Symposium on Algorithms, ESA*, volume 57 of *LIPIcs*, pages 50:1–50:18, 2016.

[107] H. Imai and M. Iri. Polygonal approximations of a curve – formulations and algorithms. In G. Toussaint, editor, *Computational Morphology*, pages 71–86. North-Holland, Amsterdam, 1988.

[108] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[109] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63(4):512–530, 2001.

[110] P. Indyk. *High-dimensional Computational Geometry*. PhD thesis, Stanford University, 2000.

[111] P. Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 10–33, 2001.

[112] P. Indyk. Approximate nearest neighbor algorithms for Fréchet distance via product metrics. In *Proceedings of the 18th ACM Symposium on Computational Geometry, SoCG*, pages 102–106, 2002.

[113] P. Indyk and J. Matoušek. Low-distortion embeddings of finite metric spaces. In J. E. Goodman and J. O'Rourke, editors, *Handbook of Discrete and Computational Geometry*, pages 177–196. CRC Press, 2004.

[114] J. Jacques and C. Preda. Functional data clustering: a survey. *Advances in Data Analysis and Classification*, pages 1–25, 2013.

[115] K. Jain, M. Mahdian, and A. Saberi. A new greedy approach for facility location problems. In *Proceedings of the 34th ACM Symposium on Theory of Computing, STOC*, pages 731–740, 2002.

[116] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary mathematics*, 26(189–206):1, 1984.

[117] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry: Theory and Applications*, 28(2-3):89–112, 2004. Previously appeared in the 18th ACM Symposium on Computational Geometry, SoCG 2002.

[118] E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery*, 7(4):349–371, 2003.

[119] E. J. Keogh and C. A. Ratanamahatana. Exact indexing of dynamic time warping. *Knowledge and information systems*, 7(3):358–386, 2005.

[120] S. G. Kolliopoulos and S. Rao. A nearly linear-time approximation scheme for the Euclidean k-median problem. *SIAM Journal on Computing*, 37(3):757–782, 2007. Previously appeared in the 7th Annual European Symposium on Algorithms, ESA 1999.

[121] D. Kotsakos, G. Trajcevski, D. Gunopulos, and C. C. Aggarwal. Time-series data clustering. In C. C. Aggarwal and C. K. Reddy, editors, *Data Clustering: Algorithms and Applications*, pages 357–379. CRC Press, 2013.

[122] Z. Kovacs-Vajna. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1266–1276, 2000.

[123] J. Krarup and S. Vajda. On Torricelli's geometrical solution to a problem of Fermat. *IMA Journal of Management Mathematics*, 8(3):215–224, 1997.

[124] I. Kremer, N. Nisan, and D. Ron. On randomized one-round communication complexity. *Computational Complexity*, 8(1):21–49, 1999.

[125] A. Krivošija and A. Munteanu. Probabilistic smallest enclosing ball in high dimensions via subgradient sampling. In G. Barequet and Y. Wang, editors, *Proceedings of the 35th International Symposium on Computational Geometry, SoCG*, volume 129 of *LIPIcs*, pages 47:1–47:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2019.

[126] A. Kumar, Y. Sabharwal, and S. Sen. Linear-time approximation schemes for clustering problems in any dimensions. *Journal of the Association for Computing Machinery*, 57(2):5:1–5:32, 2010. Previously appeared in the 32nd International Colloquium on Automata, Languages, and Programming, ICALP 2005.

[127] W. Kuszmaul. Dynamic time warping in strongly subquadratic time: Algorithms for the low-distance regime and approximate evaluation. In C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi, editors, *Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 80:1–80:15, 2019.

[128] C. Lammersen, M. Schmidt, and C. Sohler. Probabilistic k-median clustering in data streams. *Theory of Computing Systems*, 56(1):251–290, 2015. Previously appeared in the 10th International Workshop on Approximation and Online Algorithms, WAOA 2012.

[129] B. Legrand, C. Chang, S. Ong, S.-Y. Neo, and N. Palanisamy. Chromosome classification using dynamic time warping. *Pattern Recognition Letters*, 29(3):215–222, 2008.

[130] S. Li and O. Svensson. Approximating k-median via pseudo-approximation. *SIAM Journal on Computing*, 45(2):530–547, 2016. Previously appeared in the 45th ACM Symposium on Theory of Computing, STOC 2013.

[131] T. W. Liao. Clustering of time series data – a survey. *Pattern Recognition*, 38(11):1857–1874, 2005.

[132] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

[133] D. Lokshtanov, D. Marx, and S. Saurabh. Slightly superexponential parameterized problems. *SIAM Journal on Computing*, 47(3):675–702, 2018. Previously appeared in the 22nd ACM-SIAM Symposium on Discrete Algorithms, SODA 2011.

[134] J. Matoušek. Bi-lipschitz embeddings into low-dimensional euclidean spaces. *Commentationes Mathematicae Universitatis Carolinae*, 31(3):589–600, 1990.

[135] J. Matoušek. On the distortion required for embedding finite metric spaces into normed spaces. *Israel Journal of Mathematics*, 93(1):333–344, 1996.

[136] J. Matoušek. *Lectures on Discrete Geometry*. Springer-Verlag, 2002.

[137] N. Megiddo. Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems. *SIAM Journal on Computing*, 12(4):759–776, 1983.

[138] N. Megiddo and K. J. Supowit. On the complexity of some common geometric location problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

[139] S. Meintrup, A. Munteanu, and D. Rohde. Random projections and sampling algorithms for clustering of high-dimensional polygonal curves. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. B. Fox, and R. Garnett, editors, *Proceedings of the 32nd Annual Conference and Workshop on Neural Information Processing Systems, NeurIPS*, pages 12807–12817, 2019.

[140] R. R. Mettu and C. G. Plaxton. Optimal time bounds for approximate clustering. *Machine Learning*, 56(1-3):35–60, 2004. Previously appeared in the 18th Conference in Uncertainty in Artificial Intelligence, UAI 2002.

[141] M. Mitzenmacher and E. Upfal. *Probability and computing: randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[142] M. Müller. Dynamic time warping. In *Information Retrieval for Music and Motion*, pages 69–84. Springer Berlin Heidelberg, 2007.

[143] A. Munteanu. *On large-scale probabilistic and statistical data analysis*. PhD thesis, TU Dortmund University, 2018.

[144] A. Munteanu and C. Schwiegelshohn. Coresets-methods and history: A theoreticians design pattern for approximation and streaming algorithms. *Künstliche Intelligenz*, 32(1):37–53, 2018.

[145] A. Munteanu, C. Sohler, and D. Feldman. Smallest enclosing ball for probabilistic data. In *Proceedings of the 30th ACM Symposium on Computational Geometry, SoCG*, pages 214–223, 2014.

[146] A. Nath and E. Taylor. k-Median Clustering Under Discrete Fréchet and Hausdorff Distances. In S. Cabello and D. Z. Chen, editors, *Proceedings of the 36th International Symposium on Computational Geometry, SoCG*, volume 164 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:15. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020.

[147] A. Nayyeri and B. Raichel. Reality distortion: Exact and approximate algorithms for embedding into the line. In V. Guruswami, editor, *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 729–747, 2015.

[148] Y. Nesterov. *Introductory Lectures on Convex Optimization: A Basic Course*. Applied Optimization. Springer, New York, 2004.

[149] R. Pagh, F. Silvestri, J. Sivertsen, and M. Skala. Approximate furthest neighbor with application to annulus query. *Information Systems*, 64:152–162, 2017.

[150] C. H. Papadimitriou. Worst-case and probabilistic analysis of a geometric location problem. *SIAM Journal on Computing*, 10(3):542–557, 1981.

[151] M. Parizeau and R. Plamondon. A comparative analysis of regional correlation, dynamic time warping, and skeletal tree matching for signature verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):710–717, 1990.

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

[153] K. B. Pratt and E. Fink. Search for patterns in compressed time series. *International Journal of Image and Graphics*, 2(1):89–106, 2002.

[154] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972.

[155] C. E. Rasmussen and C. K. I. Williams. *Gaussian processes for machine learning*. Adaptive computation and machine learning. MIT Press, 2006.

[156] S. Ray and B. Mallick. Functional clustering by Bayesian wavelet methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68:305–332, 2006.

[157] H. L. Royden and P. M. Fitzpatrick. *Real analysis*. Prentice Hall, Boston, Mass., 4th, international edition, 2010.

[158] B. Schölkopf and A. J. Smola. *Learning with Kernels: support vector machines, regularization, optimization, and beyond*. Adaptive computation and machine learning series. MIT Press, 2002.

[159] D. R. Sheehy. Fréchet-stable signatures using persistence homology. In A. Czumaj, editor, *Proceedings of the 29th ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1100–1108. SIAM, 2018.

[160] A. Sidiropoulos, M. Bădoiu, K. Dhamdhere, A. Gupta, P. Indyk, Y. Rabinovich, H. Räcke, and R. Ravi. Approximation algorithms for low-distortion embeddings into low-dimensional spaces. *SIAM Journal on Discrete Mathematics*, 33(1):454–473, 2019. Previously appeared in the 16th ACM-SIAM Symposium on Discrete Algorithms, SODA 2005.

[161] N. Šikalo. *Development and application of a genetic algorithm-based tool for the reduction and optimization of reaction kinetic mechanisms*. PhD thesis, University of Duisburg-Essen, 2018.

[162] N. Šikalo, O. Hasemann, C. Schulz, A. Kempf, and I. Wlokas. A genetic algorithm-based method for the automatic reduction of reaction mechanisms. *International Journal of Chemical Kinetics*, 46(1):41–59, 2014.

[163] C. Sohler and D. P. Woodruff. Strong coresets for k-median and subspace approximation: Goodbye dimension. In M. Thorup, editor, *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2018*, pages 802–813, 2018.

[164] E. Sriraghavendra, K. Karthik, and C. Bhattacharyya. Fréchet distance based approach for searching online handwritten documents. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR*, volume 1, pages 461–465. IEEE, 2007.

[165] M. Stolpe, K. Bhaduri, K. Das, and K. Morik. Anomaly detection in vertically partitioned data by distributed core vector machines. In *Proceedings of Machine Learning and Knowledge Discovery in Databases - European Conference, (ECML/PKDD) Part III*, pages 321–336, 2013.

[166] D. M. J. Tax and R. P. W. Duin. Support vector data description. *Machine Learning*, 54(1):45–66, 2004.

[167] E. Terzi and P. Tsaparas. Efficient algorithms for sequence segmentation. In *Proceedings of the 6th SIAM International Conference on Data Mining, SDM*, pages 316–327. SIAM, 2006.

[168] M. Thorup. Quick k-median, k-center, and facility location for sparse graphs. *SIAM Journal on Computing*, 34(2):405–432, 2005.

[169] I. W. Tsang, J. T. Kwok, and P. Cheung. Core vector machines: Fast SVM training on very large data sets. *Journal of Machine Learning Research*, 6:363–392, 2005.

[170] TU Dortmund, ed. Regeln guter wissenschaftlicher Praxis an der TU Dortmund vom 12. Dezember 2017, 2017. also available in English: Rules of Good Scientific Practice at TU Dortmund University.

[171] M. van de Kerkhof, I. Kostitsyna, M. Löffler, M. Mirzanezhad, and C. Wenk. Global curve simplification. In M. A. Bender, O. Svensson, and G. Herman, editors, *Proceedings of the 27th Annual European Symposium on Algorithms, ESA*, volume 144 of *LIPIcs*, pages 67:1–67:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[172] M. J. van Kreveld, M. Löffler, and L. Wiratma. On optimal polyline simplification using the Hausdorff and Fréchet distance. *Journal of Computational Geometry*, 11(1):1–25, 2020. Previously appeared in the 34th International Symposium on Computational Geometry, SoCG 2018.

[173] T. K. Vintsyuk. Speech discrimination by dynamic programming. *Cybernetics*, 4(1):52–57, 1968.

[174] E. Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43(2):355–386, 1937.

[175] E. Weiszfeld. On the point for which the sum of the distances to n given points is minimum. *Annals of Operations Research*, 167:7–41, 2009. Translated from the French original and annotated by Frank Plastria.

[176] T. Wylie and B. Zhu. Protein chain pair simplification under the discrete Fréchet distance. *IEEE Transactions on Bioinformatics and Computational Biology*, 10(6):1372–1383, 2013. Previously appeared in the 8th International Symposium on Bioinformatics Research and Applications 2012.

[177] Z. Zhang, P. Tang, L. Huo, and Z. Zhou. MODIS NDVI time series clustering under dynamic time warping. *International Journal of Wavelets, Multiresolution and Information Processing*, 12(05):1461011, 2014.

[178] L. Zhao and G. Shi. A trajectory clustering method based on Douglas-Peucker compression and density for marine traffic pattern recognition. *Ocean Engineering*, 172:456–467, 2019.