k-Median Clustering of Spatial Data Sequences

Dissertation

zur Erlangung des akademischen Grades eines
Doktors der Naturwissenschaften
der Ruhr-Universität Bochum
in der Fakultät für Informatik
am Lehrstuhl Theoretische Informatik

vorgelegt von
Dennis Rohde

unter der Betreuung von
Prof. Dr. Maike Buchin

Bochum, Juli 2022

Dennis Rohde
k-Median Clustering of
Spatial Data Sequences

Day of defense: 8. July 2022

Advisor: Prof. Dr. Maike Buchin

Ruhr-Universität Bochum
Fakultät für Informatik
Lehrstuhl Theoretische Informatik
Universitätsstraße 150
44801 Bochum

Abstract

In this thesis, we develop algorithms for two fundamental analysis tasks on spatial data sequences. This kind of data arises for example whenever observations of spatial nature are repeatedly measured, like a GPS sensor measures the position of a moving object. Another example is sequential data that is usually analyzed by its shape, like temperature-, pressure-, or voltage graphs.

The first task is to summarize a given set of sequences by a single sequence that aggregates the common characteristics. We extend the well-known geometric median and (q-)mean of point sets in Euclidean space to spatial data sequences. We restrict the length of the resulting aggregate sequence by a constant ℓ to obtain a compact summary of the given data. The corresponding problems are named ℓ-median and restricted (p, q)-mean, where $p \geq 1$ and $q \geq 1$ are assumed to be constants. In the restricted (p, q)-mean problem we are given n sequences of points from some metric space, not necessarily of same length, and seek to compute an aggregate sequence such that the sum of p-dynamic time warping (p-DTW) distances, each raised to the q^{th} power, from the given sequences to the aggregate sequence is minimal. The p-DTW distance between two sequences is the p^{th} root of the minimum of the sum of pairwise distances, each raised to the p^{th} power, between the elements of the sequences, when these are aligned in a monotonic fashion. Its particular strength is the ability to handle differences in the length and in the temporal properties (e.g. phase) of the data. Furthermore, it is not sensitive to outliers in the sequences, e.g. from measurement errors or noise. A shortcoming is that it is not a metric, and it is sensitive to substantial differences in sampling rates. We provide a polynomial time exact algorithm for the restricted $(2, 2)$-mean problem in Euclidean space, a near-linear time (in the number of sequences) randomized constant factor approximation algorithm for the restricted (p, p)-mean problem in any metric space, which we derandomize in the Euclidean case, and a near-linear time randomized $(1 + \varepsilon)$-approximation algorithm for the restricted $(p, 1)$-mean problem in Euclidean space.

The setting in the ℓ-median problem is similar, but q is set to one, the points come from a Euclidean space, and we measure distances using the continuous Fréchet distance. This distance measure introduces an (implicit) linear interpolation between consecutive elements of a sequence, yielding a polygonal curve. It is the maximum distance between two points on the curves, when these are optimally aligned in a monotonic fashion. We show that the ℓ-median problem is contained in $\exists\mathbb{R}$, a complexity class between NP and PSPACE. Furthermore, we devise several approximation algorithms for this problem, among them a near-linear time randomized 34-approximation algorithm and a near-linear time randomized $(1 + \varepsilon)$-approximation algorithm that returns a sequence of length up to $2\ell - 2$.

The second analysis task we study is clustering. Here, we are given a set of sequences and seek to compute a meaningful partition of the set. More precisely, we want to split the set into k disjoint subsets, the so-called clusters, such that the elements within a cluster share a common aggregate sequence and for each given sequence, the aggregate of its cluster is more similar than an aggregate of any other cluster. To compute the aggregates, we build upon our results on the ℓ-median and restricted (p, q)-mean problems. We modify an existing algorithm to approximate a generalized k-median clustering problem. In particular, this problem subsumes the (k, ℓ)-median and the (k, ℓ, p, q)-mean clustering problems, which are extensions of the ℓ-median and the restricted (p, q)-mean problems to the clustering setting. In combination with our previous results we obtain a near-linear time randomized $(1 + \varepsilon)$-approximation algorithm for (k, ℓ)-median clustering and near-linear time randomized approximation algorithms for (k, ℓ, p, q)-means clustering problem with constant and super-constant approximation factors, which provide a trade-off between solution quality and running time. Finally, we study ε-coresets for (k, ℓ)-median clustering. An

ε-coreset is a small condensate of a given large data set that captures its core properties (for the application at hand) and is meant to serve as a proxy to run an algorithm on. We prove that sub-linear size ε-coresets for (k, ℓ)-median clustering exist and provide a near-linear time randomized algorithm to compute these. We use this algorithm to further improve one of our previous algorithms for the ℓ-median problem by means of an ε-coreset.

The remainder of the thesis deals with problems that arise from point sequences of high complexity (the length of the sequence) and of point sequences in a high-dimensional ambient space. These settings have already been studied under DTW and here we are interested in point sequences in a Euclidean space under the Fréchet distance. We prove a combined multiplicative and additive error guarantee when a $(1 \pm \varepsilon)$-embedding into a lower-dimensional Euclidean space is applied to the elements of the sequences. Furthermore, we prove that the Fréchet can not be recovered up to some constant factor, when the complexity of a given sequence is reduced by a deterministic algorithm and that the Fréchet distance can not be recovered up to some factor of $(1 + \varepsilon)$, for any $\varepsilon \in [0, \sqrt{2} - 1]$, when the complexity of a given sequence is reduced by a randomized algorithm. We achieve the latter results by reducing from problems which have $\Omega(m)$ bits one-way communication complexity for sequences of length m.

Acknowledgments

This thesis is dedicated to my lovely wife, without whose support and also pushing in the right moments I would not have went this way to the end. I am also very grateful for the companionship of my dear colleagues from Dortmund and Bochum universities. Special thanks go to my coauthors who inspired me, spent their time working with me and contributed valuable work to this thesis. Last but not least, I want to profoundly thank my supervisor Maike Buchin for her guidance, support and introducing me to the beautiful field of computational geometry.

Contents

1 Introduction

> *The computer is incredibly fast, accurate, and stupid. Man is unbelievably slow, inaccurate, and brilliant. The marriage of the two is a force beyond calculation.*
>
> Leo Cherne

Compared to the history of other sciences, little time has passed since famous scientists like Babbage, Lovelace, Peirce, Turing and Gödel laid out the foundation of computer science. Following their – mostly theoretical – work, a large industry steadily developing and improving computer hardware quickly emerged. In fact, due to the vast and vivid progress in this area, computing has rapidly conquered the summit of technology. It now permeates virtually all aspects of our everyday lives.

One trend is particularly noteworthy and has in fact directed large parts of modern *theoretical* computer science: while hardware's processing speed also undergoes a steady and significant increase, storage capacity has nowadays reached a tremendous amount, which together with the growing internet and omnipresence of a variety of sensors, lead to a sheer "data collection frenzy". At some point in the recent past, the amount of *storable* data and the amount of *processable* data began to diverge. This is not only due to storage capacity growing stronger than processing speed, but also to the fact that *classic* algorithms, which traditionally aim to solve a problem exactly, usually have at least polynomial running time dependency on the input size, sometimes even exponential and worse. While polynomial time algorithms have been considered efficient some decades ago, in the current times of *Big Data* – a term coined by the rising number of data sets that can not be processed in a whole by classic algorithms in reasonable time – *near-linear* time algorithms stand in focus. In fact, only these are considered feasible in a Big Data setting.

To achieve a near-linear running time, almost always[1] sacrifices have to be made, mostly in the form of approximation or randomization trade-offs. In detail, we do not solve the problem exactly but settle for an approximate solution, or we only return an (exact) solution with high probability. A combination of both shows to be particularly successful. Sometimes this even allows us to obtain *sub-linear* algorithms. These are algorithms that do not even read their whole input – their running time dependency on the input size is below linear. Central in this work are near-linear time (randomized) approximation algorithms. In some rare fortunate cases, we even obtain sub-linear time algorithms.

Of course, designing an algorithm (e.g. by providing pseudo-code) is only half the work. The other half consists of a theoretical analysis providing either a correctness- or an approximation-, and a (worst-case) running time guarantee. Often, analyzing the aforementioned type of algorithm turns out to be more involved than analyzing a classic algorithm, which also posed a challenge to our work; if one can not prove that a designed algorithm achieves a certain approximation factor, or can even prove that the quality of the returned solution may be arbitrarily bad, the algorithm is called *heuristic*. In fact, heuristics often play an important role in practice, as these are usually trimmed to be as time- and space efficient as possible, while – only empirically

[1] Unless the structure of the input can be exploited, a well-known example for this is the famous binary search.

verified – returning solutions of sufficient quality. Generally, one can say that it is not desirable to work with a heuristic in the long term, though. This is evident through the fact that we can not always judge the quality of the returned solution and only because the algorithm worked well so far, does not mean it *always* does, unless proven otherwise. Therefore, it is desirable to develop algorithms with theoretical guarantees, which is the main motivation of this work.

Apart from designing and improving algorithms, a way to improve processability in the frame of Big Data is *data reduction*. This umbrella term is used for techniques that consist of some form of pre-processing leading to some kind of compression while maintaining accessibility to subsequent computations. A fundamental observation that led to the development of these methods is that – depending on the application at hand – massive data sets often carry highly redundant information and removing this redundancy on the one hand saves storage space while it spares processing time towards subsequent computations on the other hand. A specific technique that is based on this observation is the computation of ε-coresets. An ε-coreset is a (problem-specific) condensate of a much larger set that can serve as a proxy to run subsequent computations on. ε-coresets (approximately) maintain the core properties (for a certain application) of the original data set, are much smaller and can be computed efficiently, often in near-linear time.
Another example of data reduction is *dimension reduction*. Here, the central observation is based on properties of the space in which the data lives. In particular, many data sets consist of compound elements comprising a number of *features*. Often, these large numbers of features are not necessary to provide the underlying information (of interest) – they can be summarized to a smaller number of features, while (approximately) maintaining the information. *Random projections* represent a specific tool that are based on this observation. These are applicable when the data lives in a (high-dimensional) Euclidean space, in particular. Here, the data is projected[2] onto a random low-dimensional subspace, which constitutes a near loss-less compression in the sense that pairwise distances between the data points are preserved up to a small error.

Nowadays, the aforementioned techniques are usually combined in a pipeline. For example, one may start with a random projection to first reduce the dimension of the data set, then compute an ε-coreset to also reduce the number of elements of the data set and then run an efficient approximation algorithm on the coreset. In fact, ε-coresets and random projections comprise another important part of this work. To combine these together with efficient algorithms also requires a deep and thorough theoretical analysis of each individual part.

1.1 Spatial Data Sequences

With the steady development of computer hardware, a variety of sensors became available. One popular example are GPS (Global Positioning System) sensors. This system, which was brought to life in the 1970's, initially only available to the US military, was soon made publicly accessible. GPS sensors have long been present in planes, vessels and of course in military devices and vehicles. Now, GPS sensors are among others also present in every new mobile phone and car, in some smartwatches and even in some cameras. Data from GPS sensors is arguably the prime example of spatial data sequences, which are the object of study in this work. GPS data usually consists of sequences of positions (trajectories) in two- or three-dimensional space, comprising latitude and longitude and if necessary – for example in planes – altitude. Sometimes these are also accompanied by additional information, such as acceleration and velocity.
Interesting applications on GPS data arise from route planing and tracking for air-, land- and maritime vehicles, as well as pure tracking, for example of hikers or animal migration movements.

[2] In fact the operation is not a proper projection in recent approaches.

Tracking is also an important application on position data from other sensors. For example, the route of a watchman through a (large) facility is often captured using an RFID transponder. Another example are weather satellites and balloons, which among other data, record position data. Here, one aims to track phenomena such as hurricanes and tornadoes to better understand these and ultimately, to forecast their emergence and routes. This is increasingly important in the light of climate change and its challenges.

A very different form of spatial data sequences comes from devices that have already existed in an analog form and are now been digitized. Electrocardiograms, blood pressure-, oxygen saturation-, temperature-, air pressure-, humidity-, seismologic-, voltage-, and current sensors are only a few examples of such devices. These measure quantities of a human body or physical quantities in general, which usually results in a sequence of measurements over time, a so-called *time-series*. Historically, many forms of time-series are analyzed by looking at their shape. For example, a doctor can judge the healthiness of a patient's heart by looking at the shape of their electrocardiogram. A technician may assess the working condition of an industrial machine by looking at a temperature and pressure curve – rising temperature and pressure may indicate machine damage – and so on.
In this sense, time-series can also carry spatial information. Furthermore, it has become usual that large facilities are monitored by a plethora of sensors. Sometimes, multiple sensors are even grouped together, forming the features of a high-dimensional time series. In any case, the sheer number of such time series that arise in modern applications make it impossible to analyze them by human means. Today, we need computer aided methods to cope.

Finally, spatial data sequences do not only arise from these natural sources, they can also result from virtual ones. For example, the stock market is a traditional source of time-series, which usually undergo a shape-based analysis. Furthermore, the internet and social media yield many forms of data that can be embedded into some space; for example user's web page access over time, where every web page is assigned a numerical value in a number of categories used to characterize them.

1.1.1 Analysis Tasks

In this work, we focus on two basic data analysis tasks, which will prove to be surprisingly hard on spatial data sequences while they are somewhat easier on isolated points (but also not easy). The first task is summary/aggregation. Here, we are given a large set of spatial data sequences and want to compute one (representative) sequence that aggregates/summarizes the properties of the given sequences. This is particularly useful when one wants to extract knowledge from data that contains information of some underlying unknown phenomenon.
One example of such a setting for isolated points (or numbers) comes from statistics. Here, one is given a sample and wants to compute a *summary statistic*, for example the *mean* or *median*. They provide location information: the central tendency of the data – in a spatial setting, just this location information is of particular interest. We study a generalization of the mean and median to spatial data sequences.
We restrict the length of the representative to obtain a *compact* summary of the *common* properties of the given sequences. This is also particularly useful when the given sequences are corrupted by measurement noise or sampling artifacts, or possess particularities we do not want to be present in our representative – think of hikers taking a detour.

The second task, called *clustering*, is to group the data. Here, one is also given a large set of spatial data sequences and wants to extract a *meaningful* grouping. In this context, "meaningful" means that the elements within the same group should be similar in combination of location and shape. In fact, this task builds upon the summary task: we want to compute groups that share high similarity to a common representative sequence, while any representative of another group is less similar. This task is particularly useful when one wants to extract knowledge from data that contains information from *several* unknown underlying phenomena.

An example of such a setting for isolated points is k-median clustering. This is an extension of the geometric median, where one wants to divide the data into k groups, each summarized by a geometric median, such that every data point is closer to the median of its group than to any other group's median.

1.1.2 Choice of Modeling

To develop methods solving the aforementioned tasks, we need to frame our setting by a mathematical formalism. We choose two models for spatial data sequences and equip each of them with a measure of *dissimilarity*, which is a key ingredient to solve the tasks.

The simplest model are point sequences over some arbitrary metric space. The specific choice of metric space is left to the application at hand, which is not part of our work. We only require the sequences elements to have metric properties.

We measure dissimilarity using the *dynamic time warping distance*. This is the sum of element-wise distances when the sequences are (monotonically) aligned in an optimal way. A particular strength of this measure is that differences in length and in temporal properties, such as phase, can be compensated. Furthermore, it is robust towards outliers within a sequence, stemming e.g. from measurement errors. A shortcoming is that it is prone to substantial differences in sampling rates.

The second and more involved model are polygonal curves in the Euclidean space. These are applicable, whenever the sequences elements can be represented by points in Euclidean space. One advantage of this approach is that we (implicitly) introduce a linear interpolation between consecutive points of a sequence. This allows to compensate arbitrary differences in sampling rates and is a reasonable assumption in many cases – measured quantities usually exhibit a smooth graph. Recall that the concept of differentiability is build on linear approximation and in turn, *smoothness* results from differentiability.

We measure dissimilarity using the *Fréchet distance*. A particular strength of this distance measure is that it takes into account the whole course of the curves, not only the pairwise distances among their vertices (the points from the sequences). We note that this behavior is intrinsic, therefore we do not need a specialized representation, we rather save the vertices only, that is, the original sequences. A shortcoming is that the Fréchet distance is very sensitive to the curve's shapes. Outliers within a sequence may drastically change the outcome.

1.2 Outline and Results

This remainder of this thesis is structured into four chapters as follows. As a general rule, every chapter that presents new results also contains thorough reviews of the related literature.

Chapter 2 Here, in a self-contained manner, we introduce all mathematical concepts used throughout the thesis. We start with mathematical basics that allow a clean and complete definition of all necessary geometric concepts and building upon these, we define the geometry fundamentals of spatial data sequences. We introduce the basics of probability theory and range spaces used to analyze the correctness of our randomized algorithms. Also, we introduce the underlying models of computation, which are necessary to analyze the running time complexity of our algorithms. Finally, we present state-of-the-art work on computing the dynamic time warping and Fréchet distances and on computing simplified sequences under these measures, that is, sequences of smaller length that are as similar as possible to their original counterparts.

Chapter 3 We study the problems of computing a median polygonal curve, respectively mean point sequence, under the Fréchet, respectively dynamic time warping distance, of restricted complexity (number of vertices/length). First, we motivate our extensions of the geometric median to point sequences by its favorable properties. Then, we study the complexity of exactly computing a median polygonal curve. Following, we devise approximation algorithms for this problem. Our main result is a $(1 + \varepsilon)$-approximation algorithm that runs in time linear in the number of given curves and polynomial in their maximum complexity.
Next, we study the problem of computing variants of a mean point sequence under the dynamic time warping distance. We show that – for a particular variant – an optimal mean point sequence in the Euclidean space can be computed in polynomial time, when the length of the mean sequence is upper bounded by a constant. We call this the *restricted* problem. Finally, we devise randomized approximation algorithms for some variants of the restricted mean sequence problem that run in near-linear time (in the number of given sequences). Our main results are a constant factor approximation algorithm for point sequences over arbitrary metric spaces, which can even be derandomized for point sequences over the Euclidean space, and a $(1 + \varepsilon)$-approximation algorithm for a certain variant and sequences in the Euclidean space.

Chapter 4 We study the k-median and k-means clustering problem for polygonal curves and point sequences, respectively. We start by adapting a k-median $(1 + \varepsilon)$-approximation algorithm from the literature and show that under our modification, the algorithm can be used to approximate any k-median problem that fits a very general definition. In particular, it can be used to approximate a k-median problem for polygonal curves under the Fréchet distance and a k-means problem for point sequences under the dynamic time warping distance, as this problem can be phrased as a k-median problem under powers of the dissimilarity measure. Using our results from Chapter 3, our main results here are a randomized $(1 + \varepsilon)$-approximation algorithm for the k-median problem for polygonal curves under the Fréchet distance and a constant factor approximation algorithm, as well as an algorithm with a large approximation factor (depending on the maximum length of a given sequence), for the k-means clustering problem for point sequences over an arbitrary metric space under the dynamic time warping distance. All algorithms run in near-linear time (in the number of input elements).
Next, we study ε-coresets for the general k-median clustering problem that our modified algorithm approximates, under the restriction that the input and cluster centers come from a metric space.

We modify a variant of the sensitivity sampling framework, which has been developed by several authors and been extended and improved throughout several works in the literature. The vanilla framework is not applicable to polygonal curves under the Fréchet distance (which form a metric space), but under our modifications we can use related work on the VC (Vapnik-Chervonenkis) dimension of metric balls under the Fréchet distance, which enables the application. Our main result is a near-linear time algorithm to compute ε-coresets of size nearly independent on the size of the original set (the dependence is only logarithmic). Finally, we use the ε-coreset result to improve the running time dependency on the input size of a median approximation algorithm from Chapter 3.

Chapter 5 Here, we focus on methods to reduce the dimension of high-dimensional polygonal curves as well as the complexity of high-complexity polygonal curves. First, we study dimension reduction. We introduce Johnson-Lindenstrauss embeddings, which can be used to embed a set of points in high-dimensional Euclidean space into a lower-dimensional subspace. We give a deep analysis bounding the distortion of the Fréchet distances among a set of polygonal curves in high-dimensional Euclidean space, when the vertices of the curves are embedded using a Johnson-Lindenstrauss embedding. Our result here is that the distortion is slightly worse than the distortion of the inter-point distances, guaranteed by the Johnson-Lindenstrauss embedding. Namely, it guarantees a distortion of the distances among the points by a factor of at most $(1 \pm \varepsilon)$. The Fréchet distance is also distorted by a factor of at most $(1 \pm \varepsilon)$, but also up to an additive of $\pm\varepsilon\alpha$, where α is the length of the maximum length line segment occurring in one of the curves. Finally, we provide experimental results showing that our embedding yields a reasonable error on real world data. In fact, the empirical distortion in any case is below a factor of $(1 \pm \varepsilon)$, without additive.
Following, we study the problem of reducing the complexity of a high-complexity polygonal curve. We use the tools of communication complexity and prove that no deterministic algorithm may compress a polygonal curve, such that the Fréchet distance to any other polygonal curve can be recovered up to some constant factor. In particular, this applies to well-known simplification algorithms. Furthermore, we prove that no randomized approximation algorithm may compress a polygonal curve, such that the Fréchet distance to any other polygonal curve can be recovered up to a factor of $(1 + \varepsilon)$ for any $\varepsilon > 0$. In fact, we only rule out the interval $\varepsilon \in [0, \sqrt{2} - 1]$. A constant factor approximation remains possible in this setting.

1.3 Publications

This manuscript is based on the following publications, which are joint work and where the authors are listed in alphabetic order. Those parts of the publications that I did not contribute to are either omitted from this thesis or only presented briefly and marked accordingly.

- Stefan Meintrup, Alexander Munteanu, and Dennis Rohde. Random Projections and Sampling Algorithms for Clustering of High-Dimensional Polygonal Curves. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS, December 8-14, Vancouver, BC, Canada*, pages 12807–12817, 2019
- Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k, ℓ)-Median Clustering for Polygonal Curves. In Dániel Marx, editor, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA, Virtual Conference, January 10 - 131*, pages 2697–2717. SIAM, 2021
- Maike Buchin and Dennis Rohde. Coresets for (k, ℓ)-Median Clustering Under the Fréchet Distance. In Niranjan Balachandran and R. Inkulu, editors, *Algorithms and Discrete Applied Mathematics - 8th International Conference, CALDAM, Puducherry, India, February 10-12, Proceedings*, volume 13179 of *Lecture Notes in Computer Science*, pages 167–180. Springer, 2022
- Maike Buchin, Anne Driemel, Koen van Greevenbroek, Ioannis Psarros, and Dennis Rohde. Approximating Length-Restricted Means under Dynamic Time Warping. *CoRR*, abs/2112.00408, 2021. (to be published)

2 Preliminaries

> *Besides it is an error to believe that rigor is the enemy of simplicity. On the contrary we find it confirmed by numerous examples that the rigorous method is at the same time the simpler and the more easily comprehended. The very effort for rigor forces us to find out simpler methods of proof.*
>
> David Hilbert

We start with some basic definitions and notations that we will use throughout the thesis. By log we denote the binary logarithm. By $\mathbb{Z}$ we denote the integers, by $\mathbb{N}$ we denote the positive integers, by $\mathbb{Q}$ we denote the rationals and by $\mathbb{R}$ we denote the reals. For $n \in \mathbb{N}$ we define $[n] = \{1, \dots, n\}$ for brevity. Furthermore, for a closed logical formula Ψ we define by $\mathbb{1}(\Psi)$ the **indicator function**, that is, the function that is 1 if Ψ is true and 0 otherwise. Let X, Y be sets and $f \colon X \to Y$ be a function. For a subset $Z \subseteq X$ we denote by $f|_Z \colon Z \to Y, z \mapsto f(z)$ the **restriction** of f to Z. For a superset $W \supset X$ we call any function $g \colon W \to Y$ with $f(x) = g(x)$ for all $x \in X$ an **extension** of f. In this work use the following asymptotic notation:

Definition 2.0.1 [235] *Let $f \colon \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ be a function. We define*

$$
\begin{aligned}
O(f) &= \{g \colon \mathbb{R}_{>0} \to \mathbb{R}_{>0} \mid \exists c, x_0 \in \mathbb{R}_{>0} \forall x \in \mathbb{R}_{\geq x_0} : g(x) \leq c \cdot f(x)\},\\
\Omega(f) &= \{g \colon \mathbb{R}_{>0} \to \mathbb{R}_{>0} \mid \exists c, x_0 \in \mathbb{R}_{>0} \forall x \in \mathbb{R}_{\geq x_0} : g(x) \geq c \cdot f(x)\},\\
\Theta(f) &= O(f) \cap \Omega(f).
\end{aligned}
$$

Next we give some basic definitions from algebra. These allow a clean definition of the Euclidean space, which is central in this thesis.

2.1 Algebra Basics

One of the most fundamental algebraic structures are groups.

Definition 2.1.1 [226, 124] *A **group** is a pair $(G, *)$ of a non-empty set G together with a binary **operation** $* \colon G \times G \to G, (g, h) \mapsto g * h$, that satisfies the following conditions:*

- ***associativity**: $(a * b) * c = a * (b * c)$ for all $a, b, c \in G$,*
- ***identity**: there exists an element $e \in G$, the so-called identity element, with $a * e = e * a = a$ for each $a \in G$ and*
- ***inverses**: for each $a \in G$ there exists an inverse element $a^{-1} \in G$ with $a * a^{-1} = a^{-1} * a = e$.*

*If also $a * b = b * a$ for all $a, b \in G$, we call the group **commutative** or **abelian**.*

Arguably the most important group is the **symmetric group**. The symmetric group $\mathcal{S}_X$ on a set X is the set of permutations (bijections) $\sigma \colon X \to X$ together with function composition as operation. By $\mathcal{S}_n$ we denote the symmetric group of degree n, i.e., the symmetric group on $[n]$.

Definition 2.1.2 [124, 35] *A (left)* ***group action*** *of a group $(G, *)$, with identity element e, on a set X is a function $\triangleright \colon G \times X \to X, (g, x) \mapsto g \triangleright x$ such that for all $x \in X$ and $g, h \in G$:*

- *$e \triangleright x = x$ and*
- *$(g * h) \triangleright x = g \triangleright (h \triangleright x)$.*

We say that the group ***acts*** *(on the left) on X (by $\triangleright$).*

Let X be a set and $(G, *)$ be a group that acts on X by some group action $\triangleright$. It is a well-known fact [124], that the action $\sigma_g \colon x \mapsto g \triangleright x$ of each element $g \in G$ is a permutation of X, thus $\sigma_g \in \mathcal{S}_X$ for all $g \in G$. Therefore, groups always act by permuting the set on which they act. This concept is vastly important and extensively used in geometry. Here, we focus on certain types of actions. First, we want the action of each element of the group, except for the action of the identity element, to be a proper permutation.

Definition 2.1.3 [35] *A group action $\triangleright$ is* ***faithful****, if $g \triangleright x = x$ for all $x \in X$ implies $g = e$.*

Furthermore, for each two elements of the set there shall exist an element of the group whose action relates them.

Definition 2.1.4 [35] *A group action $\triangleright$ is* ***transitive****, if for all $x, y \in X$ there exists a $g \in G$ with $g \triangleright x = y$.*

Large parts of modern geometry are based on the concepts of linear algebra.

2.1.1 Linear Algebra

Central objects of study in linear algebra are vector spaces and many geometric spaces, like the Euclidean space, are extensions of these.

Definition 2.1.5 [226] *A* ***real vector space*** *is a triple $(V, +, \cdot)$ of a non-empty set V, whose elements are called vectors, together with the operations $+ \colon V \times V \to V$, called* ***vector addition****, and $\cdot \colon \mathbb{R} \times V \to V$, called* ***scalar multiplication****, that satisfy the following conditions:*

- ***associativity of addition****: $u + (v + w) = (u + v) + w$ for all vectors $u, v, w \in V$,*
- ***commutativity of addition****: $u + v = v + u$ for all vectors $u, v \in V$,*
- ***identity of addition****: there exists a* ***zero vector*** *$0 \in V$, such that $0 + v = v + 0 = v$ for all vectors $v \in V$,*
- ***inverses of addition****: for each vector $v \in V$ there exists a vector in V, denoted by $-v$, with $v + -v = -v + v = 0$,*
- ***scalar identity****: $1 \cdot v = v$ for all vectors $v \in V$,*
- ***scalar compatibility****: $a(b \cdot v) = (ab) \cdot v$ for all scalars $a, b \in \mathbb{R}$ and all vectors $v \in V$ and*
- ***distributivity****: $(a + b) \cdot v = a \cdot v + b \cdot v$ and $a \cdot (v + w) = a \cdot v + a \cdot w$ for all scalars $a, b \in \mathbb{R}$ and vectors $v, w \in V$.*

From the above definition it follows that $(V, +)$ is a group. We call this group the **additive group** of the vector space. Also, we call the elements of $\mathbb{R}$ **scalars**. Usually, parts of $(V, +, \cdot)$ are vector spaces of their own. We call them subspaces.

Definition 2.1.6 [226] *A* ***subspace*** *of* $(V, +, \cdot)$ *is a vector space* $(S, +|_{S\times S}, \cdot|_{\mathbb{R}\times S})$ *with* $S \subseteq V$*. It is a* ***proper subspace****, if* $S \subset V$*. A* ***complement*** *to the subspace* $(S, +|_{S\times S}, \cdot|_{\mathbb{R}\times S})$ *is a subspace* $(T, +|_{T\times T}, \cdot|_{\mathbb{R}\times T})$*, such that* $V = S + T = \{v + w \mid v \in S, w \in T\}$ *and* $S \cap T = \{0\}$*.*

We call S and T **complementary subspaces**. Combinations of vectors under the operations of their vector space play a central role.

Definition 2.1.7 [226] *The* ***linear combination*** *of vectors* $v_1, \ldots, v_n \in V$ *with* ***coefficients*** $c_1, \ldots, c_n \in \mathbb{R}$ *is the vector* $w = c_1 \cdot v_1 + \cdots + c_n \cdot v_n$*.*

Each non-empty set of vectors generates a subspace by linear combinations.

Definition 2.1.8 [226] *The subspace* $(S, +|_{S\times S}, \cdot|_{S\times S})$ *that is* ***spanned*** *by a non-empty set* $W \subseteq V$ *of vectors is determined by the set of all linear combinations of the vectors in* W*:*

$$S = \{c_1 \cdot v_1 + \cdots + c_n \cdot v_n \mid v_1, \ldots, v_n \in W, c_1, \ldots, c_n \in \mathbb{R}\}.$$

If a proper subset of a set of vectors spans the same subspace as the set itself, then at least one vector is in a sense not necessary. We call such a vector dependent.

Definition 2.1.9 [226] *A non-empty set* $W \subseteq V$ *of vectors is* ***linearly independent****, if for all distinct* $v_1, \ldots, v_n \in W$ *we have* $0 = c_1 \cdot v_1 + \cdots + c_n \cdot v_n$ *only if* $c_i = 0$ *for all* $i \in [n]$*.*

If the vectors are not linearly independent, we call them **linearly dependent**.

Definition 2.1.10 [226] *A* ***basis*** *of* $(V, +, \cdot)$ *is a non-empty set* $B \subseteq V$ *of vectors that is linearly independent and spans* $(V, +, \cdot)$*.*

It is well-known [226] that each vector space has at least one basis and all bases of a vector space have equal cardinality. Furthermore, each vector $v \in V$ can be written as a unique linear combination of any basis B, i.e., there are unique distinct $b_1, \ldots, b_n \in B$ and unique $c_1, \ldots, c_n \in \mathbb{R}$, such that $v = c_1 \cdot v_1 + \cdots + c_n \cdot v_n$. The following concept is a notion of expressiveness of the vector space.

Definition 2.1.11 [226] *The* ***dimension*** *of* $(V, +, \cdot)$ *is the cardinality of its bases.*

It is sometimes useful to equip a basis with an order.

Definition 2.1.12 [226] *An* ***ordered basis*** *of* $(V, +, \cdot)$ *is a tuple* $B = (b_1, \ldots, b_d)$*, where* $\{b_1, \ldots, b_d\}$ *is a basis of* $(V, +, \cdot)$*. The* ***coordinate map*** *with respect to* B *is*

$$\phi_B \colon V \to \mathbb{R}^d, v = c_1 \cdot b_1 + \cdots + c_d \cdot b_d \mapsto (c_1, \ldots, c_d).$$

Note that since all vectors are unique linear combinations of the basis vectors, the coordinate map is a well-defined and bijective function. It has another useful property, which we now define.

Definition 2.1.13 [226] *A function $\rho\colon V \to W$ from a real vector space $(V, +, \cdot)$ to a real vector space $(W, +, \cdot)$ is a* ***linear transformation****, if for all $v, w \in V$ and $a, b \in \mathbb{R}$ we have*

$$\rho(a \cdot v + b \cdot w) = a \cdot \rho(v) + b \cdot \rho(w).$$

If ρ is bijective, we call it ***isomorphism****.*

If there exists an isomorphism between two vector spaces we call them **isomorphic**. Isomorphic vector spaces can be identified with each other, since they carry the same structure.

We define a popular and important real vector space, which will be used extensively in this work.

Definition 2.1.14 [226] *The* ***real coordinate space*** *is the vector space $(\mathbb{R}^d, +, \cdot)$ with*

$$+\colon \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^d, (v, w) \mapsto v + w, \qquad \cdot\colon \mathbb{R} \times \mathbb{R}^d, (a, v) \mapsto a \cdot v,$$

defined by $(v_1, \dots, v_d) + (w_1, \dots, w_d) = (v_1 + w_1, \dots, v_d + w_d)$ and $a \cdot (v_1, \dots, v_d) = (a \cdot v_1, \dots, a \cdot v_d)$. The standard basis of this space is $(e_1, \dots, e_d)$ with $e_1 = (1, 0, \dots, 0), \dots, e_d = (0, \dots, 0, 1)$.

Each d-dimensional real vector space V is isomorphic to this vector space, because for each ordered basis B of V the coordinate map ϕ_B is an isomorphism [226]. This means that there is essentially only one d-dimensional real vector space, which explains its importance. In light of geometry, we call the components (elements) of the vectors in $\mathbb{R}^d$ **Cartesian coordinates**. Working with this space, we can use the well-known tools from elementary linear algebra, such as matrices. Following the usual conventions, we regard an element of $\mathbb{R}^d$ as **column vector**.

Next, we define the basics of geometry, which are the central objects studied in this thesis.

2.2 Geometry Basics

We define the most fundamental kind of geometric spaces.

Definition 2.2.1 [226] *A* ***metric space*** *is a pair $\mathcal{X} = (X, \rho)$, where X is a non-empty set and $\rho\colon X \times X \to \mathbb{R}_{\geq 0}$ is a function that satisfies the following conditions for all $x, y, z \in X$:*

- ***identity of indiscernibles****: $\rho(x, y) = 0$ if and only if $x = y$,*
- ***symmetry****: $\rho(x, y) = \rho(y, x)$ and*
- ***triangle inequality****: $\rho(x, z) \leq \rho(x, y) + \rho(y, z)$.*

If ρ does not satisfy one or more of these conditions, we call $\mathcal{X}$ a **non-metric space**. Both kinds of spaces enable the notion of **distance** among their elements, which are generally called **points**. If $\mathcal{X}$ is a metric space, we call ρ a **metric distance function** and if $\mathcal{X}$ is a non-metric space, we call ρ a **non-metric distance function**. If it is clear from the context whether the space is metric or non-metric, we may simply call ρ **distance function**.

Of particular interest are functions between metric spaces that preserve distances.

Definition 2.2.2 [35] *An* ***isometry*** *between two metric spaces (X, μ) and (Y, ν) is a function $f\colon X \to Y$ with $\nu(f(x), f(y)) = \mu(x, y)$ for all $x, y \in X$.*

Sometimes we are interested in functions that *nearly* preserve distances.

Definition 2.2.3 [195] *A D-**embedding**, where $D \in [1, \infty)$, from a metric space (X, μ) to a metric space (Y, ν) is a function $f \colon X \to Y$ where there exists an $r \in (0, \infty)$ such that $r \cdot \mu(x, y) \leq \nu(f(x), f(y)) \leq r \cdot D \cdot \mu(x, y)$ for all $x, y \in X$. The **distortion** of f is the infimum over all $D \in [1, \infty)$ such that f is a D-embedding.*

For functions on metric spaces there exists an own notion of continuity which is based on the distance function.

Definition 2.2.4 [234] *A function $f \colon X \to Y$ between two metric spaces (X, μ) and (Y, ν) is **Lipschitz continuous**, if there exists a $k \in (0, \infty)$ such that $\nu(f(x), f(y)) \leq k \cdot \mu(x, y)$ for all $x, y \in X$.*

We say that f is k-Lipschitz. It is well-known [234] that Lipschitz continuity implies uniform continuity. We now turn to an advanced geometric space.

Definition 2.2.5 [226] *A **real normed space** is a pair $\mathcal{V} = (V, \rho)$, where V is a real vector space and $\rho \colon V \to \mathbb{R}$ is a function that satisfies the following conditions for all $x, y \in V$ and $a \in \mathbb{R}$:*

- ***positive definiteness**: $\rho(x) \geq 0$ and $\rho(x) = 0$ if and only if $x = 0$,*
- ***absolute homogeneity**: $\rho(a \cdot x) = |a| \cdot \rho(x)$ and*
- ***triangle inequality**: $\rho(x + y) \leq \rho(x) + \rho(y)$.*

We call ρ the **norm** of $\mathcal{V}$. The norm assigns to each vector $x \in X$ of the space a notion of **length**. Furthermore, it is a well-known fact [226] that a norm induces a metric distance function among the vectors of the space: $(x, y) \mapsto \rho(y - x)$. Hence, each real normed space is also a metric space. We define a closely related geometric space.

Definition 2.2.6 [226] *A **real inner product space** is a pair $\mathcal{V} = (V, \rho)$, where V is a real vector space and $\rho \colon V \times V \to \mathbb{R}$ is a function that satisfies the following conditions for all $x, y, z \in V$ and $a, b \in \mathbb{R}$:*

- ***positive definiteness**: $\rho(x, x) \geq 0$ and $\rho(x, x) = 0$ if and only if $x = 0$,*
- ***symmetry**: $\rho(x, y) = \rho(y, x)$ and*
- ***linearity**: $\rho(a \cdot x + b \cdot y, z) \leq a \cdot \rho(x, z) + b \cdot \rho(y, z)$.*

We call ρ the **inner product** of $\mathcal{V}$. An inner product enables the notion of **angle** between any two vectors of the space. Also, it is a well-known fact [226] that an inner product induces a norm: $x \mapsto \sqrt{\rho(x, x)}$. Therefore, each real inner product space is also a real normed space and thus also a metric space. Furthermore, note that for each subspace S of V there exists a unique complementary subspace, denoted by $S^{\perp}$, with vectors $\{v \in V \mid \forall w \in S : \rho(v, w) = 0\}$, called the **orthogonal complement** of S [226].

Of particular interest are linear transformations that preserve inner products.

Definition 2.2.7 [226, 35] *An **isometry** between two inner product spaces (V, μ) and (W, ν) is a linear transformation $\rho \colon V \to W$ with $\nu(\rho(x), \rho(y)) = \mu(x, y)$ for all $x, y \in V$. If ρ is an isomorphism, we call ρ an **isometric isomorphism** and if also $V = W$ we call ρ an **isometric automorphism**.*

If there exists an isometric isomorphism between two inner product spaces, we call them **isometrically isomorphic**. An interesting fact [226] is that an inner product ρ can be recovered from its induced norm $\rho'(v) = \sqrt{\rho(v,v)}$ via the identity: $\rho(v,w) = \frac{1}{4}(\rho'(v+w)^2 - \rho'(v-w)^2)$. Consequently, a linear transformation that preserves the induced norms is an isometry.

Theorem 2.2.8 [226] *Let (V,μ) and (W,ν) be real inner product spaces and $\rho\colon V \to W$ be a linear transformation. ρ is an isometry if and only if it preserves the norms induced by μ and ν, i.e., $\sqrt{\nu(\rho(v),\rho(v))} = \sqrt{\mu(v,v)}$ for all $v \in V$.*

We define a popular and important real inner product space.

Definition 2.2.9 [226, 35] *The (standard) d-dimensional **Euclidean vector space** is the real coordinate space $(\mathbb{R}^d, +, \cdot)$, equipped with the Euclidean inner product, called **dot product**, defined*

$$\langle(x_1,\dots,x_d),(y_1,\dots,y_d)\rangle = \sum_{i=1}^{d} x_i \cdot y_i,$$

*the **Euclidean norm** induced by its inner product, denoted by*

$$\|x\| = \sqrt{\langle x,x\rangle},$$

*and the **Euclidean distance** between $x, y \in \mathbb{R}^d$, induced by its norm:*

$$\|x-y\|.$$

Technically, each d-dimensional real inner product space is a Euclidean vector space and isometrically isomorphic to this space [35]. Therefore, there is essentially one d-dimensional Euclidean vector space. Many geometric spaces have an affine structure, which we now define.

Definition 2.2.10 [35] *An **affine space** is a triple $(X,Y,\triangleright)$ of a non-empty set X, a (real) vector space Y and a faithful and transitive group action $\triangleright\colon Y \times X \mapsto X$ of the additive group of Y. The **dimension** of $(X,Y,\triangleright)$ is the dimension of Y and we say that the vector space Y underlies the affine space.*

We call the elements of X **points** and the action $x \mapsto y \triangleright x$ of $y \in Y$ we call **translation** (by the vector y). We write $x + y$ instead of $y \triangleright x$. For $x_1, x_2 \in X$ we denote by $\overrightarrow{x_1x_2} = x_2 - x_1$ the vector that translates x_1 to x_2. Also note that by definition $|X| = |Y|$ [35]. The notion of subspaces naturally extends to the affine case.

Definition 2.2.11 [35] *A **subspace** of $(X,Y,\triangleright)$ is an affine space $(X',Y',\triangleright|_{Y'\times X'})$ with point set $X' = x + Y = \{x + y \mid y \in Y'\}$ generated by a point $x \in X$ and a vector subspace $Y' \subseteq Y$, which is called the **direction** of the subspace. Two subspaces are **complementary subspaces**, if their directions are complementary.*

Note that the subspace generated by any point $x \in X$ and any direction Y' is unique and that two complementary subspaces share exactly one point [35]. The subspaces generated by $x = 0$ are the *linear* subspaces.

Definition 2.2.12 [35] *A **projection** from $(X,Y,\triangleright)$ to a subspace $(X',Y',\triangleright|_{Y'\times X'})$ parallel to a complementary (with respect to Y') vector subspace $Y'' \subseteq Y$, is the function $p\colon X \to X'$ that maps $x \in X$ to the unique point shared by X' and $x + Y''$. p is an **orthogonal projection** if Y is an inner product space and Y'' is the orthogonal complement of Y'.*

We now define an old and arguably the most extensively studied geometric space, which fits our perception of the physical universe.

Definition 2.2.13 [35] *The (standard d-dimensional affine)* ***Euclidean space*** *is the affine space with point set $\mathbb{R}^d$ and underlying (standard) d-dimensional Euclidean vector space. The (metric) distance between points $x, y \in \mathbb{R}^d$ is given by the length of the vector that translates x to y.*

Note that technically any point set X with $|X| = |\mathbb{R}^d|$ yields an affine Euclidean space [35]. Furthermore, an **isometry** between two Euclidean spaces is a metric isometry [35], which is per definition of the space an isometry between the associated inner product spaces (and vice versa). Throughout the whole work, we assume d to be constant.

Despite its affine nature, we endow the Euclidean space with a distinguished reference point, called **origin**. Now, we can associate with each point p its **position vector**, i.e., the vector v that translates the origin to p and consequently we must associate the zero vector 0 with the origin. Hence, $p = 0 + v$ must hold for each point p with position vector v. If we now take $+$ to be the vector addition of $(\mathbb{R}^d, +, \cdot)$ and set $v = p$, this justifies the choice of $\mathbb{R}^d$ as point set.

In the following we will therefore no longer strictly distinguish between points and vectors and for brevity, we will from now on denote the Euclidean space by $\mathbb{R}^d$. Depending on the situation, we either view the elements of $\mathbb{R}^d$ as points, position vectors, or translations. For example, if we speak about the distance between $x, y \in \mathbb{R}^d$, then x and y are points, if we take the inner product between x and y, then x and y are position vectors and if we take the norm of $x \in \mathbb{R}^d$, then x is a position vector or a translation vector and the norm gives its length.

This intimate relationship between points, translations and position vectors reflects in the relations between Euclidean distance, Euclidean norm and Euclidean inner product. For $x, y \in \mathbb{R}^d$:

$$\|x - y\| = \sqrt{\langle x-y, x-y\rangle} = \sqrt{\langle x,x\rangle - \langle x,y\rangle - \langle y,x\rangle + \langle x,x\rangle} = \sqrt{\|x\|^2 - 2\langle x,y\rangle + \|y\|^2}. \quad \text{(PV)}$$

In this setting, the isometries of the Euclidean space are each given by a multiplication with an orthogonal matrix A followed by a translation by a vector v, i.e., $p \mapsto Ap + v$ [35]. These mappings are called (Euclidean) **motions**. Motions naturally form a group under composition, the so-called **Euclidean group**. The matrices form a (sub-)group under matrix multiplication, the so-called orthogonal group.

Definition 2.2.14 [35] *The* ***orthogonal group*** *$\mathcal{O}(d)$ is the group of real $d \times d$ matrices with determinant in $\{-1, 1\}$. The operation of the group is matrix multiplication.*

We define some important types of subsets of the Euclidean space, starting with regions. These have the property that for each two contained points there exists a path (formally a curve, see Definition 2.3.7) that connects them and that is completely contained in the region.

Definition 2.2.15 [89] *An* ***open region*** *is a subset $P \subseteq \mathbb{R}^d$ such that for all points $p = (p_1, \ldots, p_d) \in P$, there exists an $\varepsilon \in (0, \infty)$ such that all points $q = (q_1, \ldots, q_d) \in \mathbb{R}^d$ with $|p_i - q_i| < \varepsilon$ for all $i \in [d]$ are a member of P.*

A ***closed*** *region P is the union of an open region P' with its* ***boundary****, i.e., the set of points $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$ such that for all $\varepsilon \in (0, \infty)$ there exists a $q = (q_1, \ldots, q_d) \in P'$ with $|p_i - q_i| < \varepsilon$ for all $i \in [d]$.*

A related type of subsets are compact sets, which are in some sense finite.

Definition 2.2.16 [226] *A **compact set** is a set of points $P \subseteq \mathbb{R}^d$ that contains its boundary and has finite diameter, i.e.,* $\sup_{p,q \in P} \|p - q\| < \infty$.

Compact sets must not necessarily be closed regions, since they may not be connected. An important type of regions can be generated from convex combinations.

Definition 2.2.17 [226] *A **convex combination** of a sequence of points $p_1, \dots, p_n \in \mathbb{R}^d$ is a point $q = \sum_{i=1}^{n} t_i \cdot p_i$, where $t_1, \dots, t_n \in [0, 1]$ and $\sum_{i=1}^{n} t_i = 1$.*

Of particular interest are convex combinations of two points, which are named line segments, and the related objects.

Definition 2.2.18 [226, 35] *A **line** is the set of points $\{p + \lambda(q - p) \mid \lambda \in \mathbb{R}\}$, a **half-line** is the set of points $\{p + \lambda(q - p) \mid \lambda \in \mathbb{R}_{\geq 0}\}$ and a **line-segment** is the set of points $\{p + \lambda(q - p) \mid \lambda \in [0, 1]\}$, where $p, q \in \mathbb{R}^d$.*

Each of these is determined by two points p and q. A line segment between two points p, q will be denoted by $\overline{pq}$. Furthermore, for $\lambda \in [0, 1]$ we denote by $\mathrm{lp}\left(\overline{pq}, \lambda\right)$ the point $(1 - \lambda)p + \lambda q$, lying on $\overline{pq}$. Each vector $v \in \mathbb{R}^d$ also naturally determines a line and a half-line through the origin: $\{\lambda v \mid \lambda \in \mathbb{R}\}$, respectively $\{\lambda v \mid \lambda \in \mathbb{R}_{\geq 0}\}$. When there exists a line on which a given set of points lies, we say that the points are **collinear**.

We call a set of points convex if it contains all line segments determined by its points.

Definition 2.2.19 [226] *A **convex** set is a subset $X \subseteq \mathbb{R}^d$ that contains each line segment $\overline{pq}$ determined by any two points $p, q \in X$.*

If a set is not convex, we may be interested in a certain convex set that contains it, namely the convex hull of the set.

Definition 2.2.20 [226] *The **convex hull** of a set $X \subseteq \mathbb{R}^d$ is the smallest convex set $H \subseteq \mathbb{R}^d$ that contains X.*

It is a well-known fact that the convex hull of X is the set of all convex combinations of the points in X [226]. Other important convex sets are balls.

Definition 2.2.21 [35] *The (closed) **ball** of radius $r \in \mathbb{R}_{\geq 0}$ around the center point $p \in \mathbb{R}^d$, denoted by $\mathrm{B}(p, r)$, is the set $\{q \in \mathbb{R}^d \mid \|p - q\| \leq r\}$. The **sphere** of radius $r \in \mathbb{R}_{>0}$ around the center point $p \in \mathbb{R}^d$ is the boundary of the corresponding closed ball: $\{q \in \mathbb{R}^d \mid \|p - q\| = r\}$ and the corresponding **open ball** is the volume enclosed by the sphere: $\{q \in \mathbb{R}^d \mid \|p - q\| < r\}$.*

Every two distinct points on a sphere determine a circle with the same radius as the sphere. The two points divide the circle into two parts, which we call **arcs**. We now define the related concept of angles.

Definition 2.2.22 [35] *The **angle** between two vectors $x, y \in \mathbb{R}^d$, denoted by $\sphericalangle(x, y)$, is a number from $[0, \pi]$ defined by*

$$\sphericalangle(x, y) = \arccos\left(\frac{\langle x, y \rangle}{\|x\| \cdot \|y\|}\right).$$

We can picture this number as the length of the shorter arc determined by the two intersection points of the half-lines determined by x and y with the sphere of unit radius around the origin. Since this circle has circumference 2π, we obtain a number between 0 and π. The following identity is now immediate: $\langle x, y\rangle = \|x\| \cdot \|y\| \cdot \cos \sphericalangle(x, y)$. Combining this with Eq. (PV), we get the **law of cosines**: $\|x - y\|^2 = \|x\|^2 + \|y\|^2 - 2\|x\|\|y\| \cos \sphericalangle(x, y)$.

The objects we defined so far naturally mostly consist of an infinite number of points. We can cover such an object by a Euclidean grid, which is a finite set of points, such that every point from the object has a point of the grid in proximity.

Definition 2.2.23 [130] *Given a number $r \in \mathbb{R}_{>0}$, for $p = (p_1, \ldots, p_d) \in \mathbb{R}^d$ we define by*

$$G(p, r) = (\lfloor p_1/r \rfloor \cdot r, \ldots, \lfloor p_d/r \rfloor \cdot r)$$

*the r-**grid point** of p. Let $X \subseteq \mathbb{R}^d$ be a subset of $\mathbb{R}^d$. The **grid** of cell width r that covers X is the set $\mathbb{G}(X, r) = \{G(p, r) \mid p \in X\}$.*

Such a grid partitions the set X into cubic regions and for each $r \in \mathbb{R}_{>0}$ and $p \in X$ we have that $\|p - G(p, r)\| \leq \sqrt{d}r$. We now turn to the geometric spaces in the focus of this work.

2.3 Geometry of Spatial Data Sequences

When dealing with sequences of spatial data, one typically assumes that the elements of the sequences themselves stem from some metric space, and to measure distances among the sequences one relies on the distances among their elements. In fact, when comparing two sequences, there are numerous possibilities to combine the distances among their elements to a single number, representing their distance. In any case, we take into account a finite number of underlying distances, resulting in a so-called **discrete distance function**. However, if we introduce some kind of interpolation between every two consecutive elements of a sequence – for example a linear interpolation by connecting them with the line segment that they determine, thereby obtaining polygonal curves –, then we can take into account an uncountably infinite number of underlying distances. This results in a so-called **continuous distance function**. Whether an interpolation is desired or not depends on the specific type of data and the application at hand.

2.3.1 Point Sequences in Metric Spaces

If we do not assume any kind of interpolation we are dealing with sequences of points from some metric space, which we simply call point sequences.

Definition 2.3.1 *A **point sequence** over a metric space (M, ϑ) is a tuple $(\sigma_1, \ldots, \sigma_m) \in M^m$, where $m \in \mathbb{N}_{>1}$ is its **complexity**, denoted by $|\sigma|$ and $\sigma_1, \ldots, \sigma_m$ are its **vertices**.*

By $M^{\leq m} = \bigcup_{i=2}^{m} M^i$ we denote the set of all point sequences of complexity at most m over (M, ϑ) and by $M^* = \bigcup_{i=2}^{\infty} M^i$ we denote the set of all point sequences over (M, ϑ). Point sequences can be merged together.

Definition 2.3.2 *The **concatenation** of a point sequence $\sigma = (\sigma_1, \ldots, \sigma_{|\sigma|})$ with a sequence $\tau = (\tau_1, \ldots, \tau_{|\tau|})$ is denoted by $\sigma \oplus \tau$ and is defined as the point sequence $(\sigma_1, \ldots, \sigma_{|\sigma|}, \tau_1, \ldots, \tau_{|\tau|})$.*

To define a notion of distance between point sequences of possibly different complexity we need to align them.

Definition 2.3.3 [227, 95] *For $m_1, m_2 \in \mathbb{N}_{>1}$, let $\mathcal{W}_{m_1,m_2}$ denote the set of all (m_1, m_2)-**warpings**, that is, the set of all sequences $(i_1, j_1), \ldots, (i_n, j_n)$ with*

- $i_1 = j_1 = 1$, $i_n = m_1$, $j_n = m_2$ *and*
- $(i_k - i_{k-1}, j_k - j_{k-1}) \in \{(0, 1), (1, 0), (1, 1)\}$ *for each* $k \in \{2, \ldots, n\}$.

In the literature, a warping is also called **coupling** or **matching**. We do not use the latter to distinguish from matchings used to define the continuous Fréchet distance. The discrete Fréchet distance yields a distance measure that is similar to the Chebyshev distance, which is the metric implied by the ℓ_∞ norm, in the sense that both are determined by a maximum over a set of underlying distances, whereas the former is over Euclidean distances and the latter over coordinate-wise distances.

Definition 2.3.4 [95] *The **discrete Fréchet distance** between $\sigma = (\sigma_1, \ldots, \sigma_{m_1}) \in M^{m_1}$ and $\tau = (\tau_1, \ldots, \tau_{m_2}) \in M^{m_2}$ is defined as*

$$\mathrm{d}_{\mathrm{dF}}(\sigma, \tau) = \min_{W \in \mathcal{W}_{m_1,m_2}} \max_{(i,j) \in W} \vartheta(\sigma_i, \tau_j).$$

In this work, we always measure the distance between two point sequences using the p-dynamic time warping distance. This distance measure is similar to the metrics implied by the ℓ_p norms.

Definition 2.3.5 [227] *For $p \in [1, \infty)$ the p-**dynamic time warping distance**, in short p-DTW, between two point sequences $\sigma = (\sigma_1, \ldots, \sigma_{m_1}) \in M^{m_1}$ and $\tau = (\tau_1, \ldots, \tau_{m_2}) \in M^{m_2}$ is defined as*

$$\mathrm{d}_{\mathrm{DTW}_p}(\sigma, \tau) = \min_{W \in \mathcal{W}_{m_1,m_2}} \left(\sum_{(i,j) \in W} \vartheta(\sigma_i, \tau_j)^p \right)^{\frac{1}{p}}.$$

We call a warping $W \in \underset{W \in \mathcal{W}_{m_1,m_2}}{\arg\min} \left(\sum_{(i,j) \in W} \vartheta(\sigma_i, \tau_j)^p \right)^{\frac{1}{p}}$ an optimal p-warping between σ and τ.

For any $m \in \mathbb{N}_{>1}$, $(M^{\leq m}, \mathrm{d}_{\mathrm{DTW}_p})$ is a non-metric space, since the p-dynamic time warping distance does not fulfill the identity of indiscernibles and the triangle inequality. However, the p-dynamic time warping distance fulfills a relaxed variant of the triangle inequality [179].

We note that there is a continuous variant of the dynamic time warping distance, but there is no algorithm to exactly compute this distance measure in general. However, there is an $(1 + \varepsilon)$-approximation algorithm with running time $O(\alpha^4 m^4 / \varepsilon^2 \log \alpha m / \varepsilon)$ [193] for point sequences from $\left(\mathbb{R}^d\right)^{\leq m}$, where α is the maximum ratio of the distances between two vertices from both sequences and a more practical additive approximation algorithm [44] for point sequences over $\mathbb{R}^d$. Furthermore, very recently a polynomial time exact algorithm for point sequences over $\mathbb{R}$ was published [62].

Weak Triangle Inequality

As we already mentioned, the dynamic time warping distance fulfills a loose variant of the triangle inequality. In particular, Lemire [179] shows that given $\tau_1, \tau_2, \tau_3 \in M^m$, and $p \in [1, \infty)$, we have $\mathrm{d}_{\mathrm{DTW}_p}(\tau_1, \tau_3) \leq m^{1/p} \cdot (\mathrm{d}_{\mathrm{DTW}_p}(\tau_1, \tau_2) + \mathrm{d}_{\mathrm{DTW}_p}(\tau_2, \tau_3))$.

We slightly generalize the above inequality in a way that implies a better bound for the distance between two short point sequences using the distances to a potentially longer point sequence.

Lemma 2.3.6 *For any $m_1, m_2 \in \mathbb{N}$, let $\sigma = (\sigma_1, \ldots, \sigma_{|\sigma|}), \upsilon = (\upsilon_1, \ldots, \upsilon_{|\upsilon|}) \in M^{\leq m_1}$, $\tau = (\sigma, \ldots, \tau_{m_2}) \in M^{m_2}$, and $p \in [1, \infty)$. It holds that*

$$\mathrm{d}_{\mathrm{DTW}_p}(\sigma, \upsilon) \leq m_1^{1/p} \cdot (\mathrm{d}_{\mathrm{DTW}_p}(\sigma, \tau) + \mathrm{d}_{\mathrm{DTW}_p}(\tau, \upsilon)) \,.$$

Proof. Let $W_{\sigma\upsilon} \in \mathcal{W}_{|\sigma|,|\upsilon|}$ be an optimal warping between σ and υ. Let $W_{\sigma\tau} \in \mathcal{W}_{|\sigma|,|\tau|}$ be an optimal warping between σ and τ, and $W_{\tau\upsilon} \in \mathcal{W}_{\tau,\upsilon}$ be an optimal warping between τ and υ. Let

$$S_{\sigma\upsilon} = \{(i,k,j) \in [|\sigma|] \times [|\tau|] \times [|\upsilon|] \mid (i,k) \in W_{\sigma\tau}, (k,j) \in W_{\tau\upsilon}\}$$

and

$$W'_{\sigma\upsilon} = \{(i,j) \in [|\sigma|] \times [|\upsilon|] \mid \exists k \in [|\tau|] : (i,k,j) \in S_{\sigma\upsilon}\}.$$

Then,

$$\begin{aligned}
\mathrm{d}_{\mathrm{DTW}_p}(\sigma, \upsilon) = \left(\sum_{(i,j)\in W_{\sigma\upsilon}} \vartheta(\sigma_i, \upsilon_j)^p\right)^{1/p} &\leq \left(\sum_{(i,j)\in W'_{\sigma\upsilon}} \vartheta(\sigma_i, \upsilon_j)^p\right)^{1/p} \\
&\leq \left(\sum_{(i,k,j)\in S_{\sigma\upsilon}} (\vartheta(\sigma_i, \tau_k) + \vartheta(\tau_k, \upsilon_j))^p\right)^{1/p} \\
&\leq \left(\sum_{(i,k,j)\in S_{\sigma\upsilon}} \vartheta(\sigma_i, \tau_k)^p\right)^{1/p} + \left(\sum_{(i,k,j)\in S_{\sigma\upsilon}} \vartheta(\tau_k, \upsilon_j)^p\right)^{1/p} \\
&\leq m_1^{1/p} \cdot \mathrm{d}_{\mathrm{DTW}_p}(\sigma, \tau) + m_1^{1/p} \cdot \mathrm{d}_{\mathrm{DTW}_p}(\tau, \upsilon),
\end{aligned}$$

where the second inequality holds by the triangle inequality and the third inequality holds by Minkowski's inequality [226]. □

2.3.2 Polygonal Curves in the Euclidean Space

If we apply a linear interpolation between each two consecutive data elements by connecting them with the line segment that they determine, we obtain polygonal curves from the sequences. We formally define curves.

Definition 2.3.7 [89, 21] *A (parameterized)* ***curve*** *is a continuous function $\tau \colon [0,1] \to \mathbb{R}^d$.*

If there exist two distinct t_1, t_2 with $\tau(t_1) = \tau(t_2)$, we say that τ self-intersects. A crucial tool when working with curves are reparameterizations, these are the continuous extensions of warpings/couplings.

Definition 2.3.8 [21] *A **reparameterization** is a continuous and bijective function $h\colon [0,1] \to [0,1]$ with $h(0) = 0$ and $h(1) = 1$. By $\mathcal{H}$ we denote the set of all reparameterizations.*

We now define polygonal curves.

Definition 2.3.9 [21] *A **polygonal curve** is a curve τ, such that there exist $h \in \mathcal{H}$, $v_1, \ldots, v_m \in \mathbb{R}^d$, no three consecutive on a line, called **vertices**, and $t_1, \ldots, t_m \in [0,1]$ with $t_1 = 0 < \cdots < t_m = 1$, called **instants**, such that*

$$\tau(h(t)) = \begin{cases} \operatorname{lp}\left(\overline{v_1 v_2}, \frac{h(t)-t_1}{t_2-t_1}\right), & \text{if } h(t) \in [0, t_2) \\ \vdots & \\ \operatorname{lp}\left(\overline{v_{m-1} v_m}, \frac{h(t)-t_{m-1}}{t_m-t_{m-1}}\right), & \text{if } h(t) \in [t_{m-1}, 1] \end{cases}.$$

In the following we will assume that h is the identity function, because the Fréchet distance, which is subsequently defined, is invariant under reparameterizations [21]. We only need h to keep our definition general. Further, we call m the **complexity** of τ, denoted by $|\tau|$, and we call the line segments $\overline{v_1 v_2}, \ldots, \overline{v_{m-1} v_m}$ the **edges** of τ. Sometimes we will argue about a **subcurve** τ of a given curve σ. We will then refer to τ by restricting the domain of σ.

Remark 2.3.10 *In Definition 2.3.9 we introduced h and the restriction of every three consecutive vertices not being collinear to avoid distinguishing between complexity and* proper *complexity. For example, let $\sigma \in \mathbb{R}^d_*$ be a curve with vertices v_1, v_2, v_3 and v_4, and instants t_1, t_2, t_3 and t_4. Assume that v_1, v_2 and v_3 are collinear, but v_1, v_2, v_3 and v_4 are not. Clearly, σ has complexity 4, but its* proper *complexity would be 3, since we can remove v_2 without changing the appearance of σ.*

Now, assume that $\frac{t_2-t_1}{t_3-t_1} > \frac{\|v_2-v_1\|}{\|v_3-v_1\|}$. In such a case we need h to compensate the parameterization of σ, such that $\sigma(h(t_2))$ is on the correct relative position on $\overline{v_2 v_3}$ and so we can use $\operatorname{lp}\left(\overline{v_1 v_3}, \frac{h(t)-t_1}{t_3-t_1}\right)$, thereby effectively removing the unnecessary vertex v_2.

The continuous Fréchet distance has been introduced more than a hundred years ago by Maurice Fréchet [114] and nearly thirty years ago was rediscovered by Alt and Godau [21], who used it for shape matching in computational geometry.

Definition 2.3.11 [114, 21] *The **Fréchet distance** between curves σ and τ is defined as*

$$\mathrm{d_F}(\sigma, \tau) \;=\; \inf_{h \in \mathcal{H}} \max_{t \in [0,1]} \|\sigma(t) - \tau(h(t))\|.$$

Sometimes, given two curves σ, τ, we will refer to an $h \in \mathcal{H}$ as **matching** between σ and τ.

Note that there must not exist a matching $h \in \mathcal{H}$, such that $\max_{t\in[0,1]} \|\sigma(t) - \tau(h(t))\| = \mathrm{d_F}(\sigma, \tau)$. This is due to the fact that in some cases a matching realizing the Fréchet distance would need to match multiple points $p_1, \ldots, p_n$ on τ to a single point q on σ, which is not possible since matchings need to be bijections, but the $p_1, \ldots, p_n$ can get matched arbitrarily close to q, realizing $\mathrm{d_F}(\sigma, \tau)$ in the limit, which we formalize in the following proposition.

Proposition 2.3.12 *Let $\sigma, \tau\colon [0,1] \to \mathbb{R}^d$ be curves. Let $r = \mathrm{d_F}(\sigma, \tau)$. There exists a sequence $(h_i)_{i=1}^{\infty}$ in $\mathcal{H}$, such that $\lim\limits_{i\to\infty} \max\limits_{t\in[0,1]} \|\sigma(t) - \tau(h_i(t))\| = r$.*

Proof. Define $\rho\colon \mathcal{H} \to \mathbb{R}_{\geq 0}, h \mapsto \max_{t\in[0,1]} \|\sigma(t) - \tau(h(t))\|$ with image $R = \{\rho(h) \mid h \in \mathcal{H}\}$. Per definition, we have $\mathrm{d_F}(\sigma, \tau) = \inf R = r$.

For any non-empty subset X of $\mathbb{R}$ that is bounded from below and for every $\varepsilon > 0$ it holds that there exists an $x \in X$ with $\inf X \leq x < \inf X + \varepsilon$, by definition of the infimum. Since $R \subseteq \mathbb{R}$ and $\inf R$ exists, for every $\varepsilon > 0$ there exists an $r' \in R$ with $\inf R \leq r' < \inf R + \varepsilon$.

Now, let $a_i = 1/i$ be a zero sequence. For every $i \in \mathbb{N}$ there exists an $r_i \in R$ with $r \leq r_i < r + a_i$, thus $\lim\limits_{i\to\infty} r_i = r$.

Let $\rho^{-1}(r') = \{h \in \mathcal{H} \mid \rho(h) = r'\}$ be the preimage of ρ. Since ρ is a function, $|\rho^{-1}(r')| \geq 1$ for each $r' \in R$. Now, for $i \in \mathbb{N}$, let h_i be an arbitrary element from $\rho^{-1}(r_i)$. By definition, it holds that

$$\lim_{i\to\infty} \max_{t\in[0,1]} \|\sigma(t) - \tau(h_i(t))\| = \lim_{i\to\infty} \rho(h_i) = \lim_{i\to\infty} r_i = r = \inf R,$$

which proves the claim. □

Now we introduce the classes of polygonal curves we are interested in. Let σ, τ be polygonal curves. We define the relation $\sigma \sim \tau \iff \exists h \in \mathcal{H} : \sigma = \tau \circ h$, which is an equivalence relation [21].

Definition 2.3.13 *For $d \in \mathbb{N}$, we define by $\mathbb{R}^d_*$ the set of equivalence classes with respect to $\sim$, of polygonal curves in ambient space $\mathbb{R}^d$, and by $\mathbb{Q}^d_*$ we define the set of equivalence classes with respect to $\sim$, of polygonal curves in ambient space $\mathbb{R}^d$, where all vertices of the curves come from $\mathbb{Q}^d$. For $m \in \mathbb{N}$ we define by $\mathbb{R}^d_m$, respectively $\mathbb{Q}^d_m$, the subclass of polygonal curves of complexity at most m.*

In the following, we identify each polygonal curve σ, as well its whole equivalence class, with the representative that is parameterized such that h in Definition 2.3.9 is the identity. Then, $(\mathbb{R}^d_*, \mathrm{d_F})$, respectively $(\mathbb{Q}^d_*, \mathrm{d_F})$, is a metric space, as is $(\mathbb{R}^d_m, \mathrm{d_F})$, respectively $(\mathbb{Q}^d_m, \mathrm{d_F})$, for any $m \in \mathbb{N}_{>1}$ [21].

2.4 Probability Theory

We define the basics of probability theory, which is the foundation of randomized algorithms and their analysis. The underlying model of each probabilistic experiment is a probability space.

Definition 2.4.1 [202, 237] *A **probability space** is a triple $(\Xi, \mathcal{E}, \Pr)$, of a non-empty set Ξ, the so-called **sample space**, a **σ-algebra** $\mathcal{E} \subseteq 2^{\Xi}$ containing the so-called **events**, and a function $\Pr\colon \mathcal{E} \to [0,1], E \mapsto \Pr[E]$, the so-called **probability function**, satisfying the following conditions:*

- $\emptyset \in \mathcal{E}$,
- $\forall E \in \mathcal{E} : (\Xi \setminus E) \in \mathcal{E}$,
- *for all sequences $(E_i)_{i=1}^{\infty}$ of events in $\mathcal{E}$ we have $(\bigcup_{i=1}^{\infty} E_i) \in \mathcal{E}$,*
- $\Pr[\Xi] = 1$ *and*
- *for all sequences $(E_i)_{i=1}^{\infty}$ in $\mathcal{E}$ of pairwise disjoint events we have $\Pr\left[\bigcup_{i=1}^{\infty} E_i\right] = \sum_{i=1}^{\infty} \Pr[E_i]$.*

The elements of Ξ are the possible outcomes of the probabilistic experiment, consequently only one can occur at a time. We call them **elementary events**. Let $E \in \mathcal{E}$ be an event. $\Pr[E]$ is the probability that one of the elementary events in E occurs and by $\overline{E} = \Xi \setminus E$ we denote the **complementary event**. From the above definition it is immediate that $\Pr[\overline{E}] = 1 - \Pr[E]$. Events can depend on one another.

Definition 2.4.2 [202] *Let $E_1, \dots, E_n \in \mathcal{E}$ be events. These are (mutually)* ***independent*** *if and only if for any $I \subseteq [n]$ we have*

$$\Pr\left[\bigcap_{i \in I} E_i\right] = \prod_{i \in I} \Pr[E_i].$$

It is often of interest whether at least one event from a collection of events occurs. The following bound on this probability is called **union bound**.

Proposition 2.4.3 [202] *Let $E_1, \dots, E_n \in \mathcal{E}$ be events. It holds that*

$$\Pr\left[\bigcup_{i=1}^{n} E_i\right] \leq \sum_{i=1}^{n} \Pr[E_i].$$

Sometimes we want to study the probability of an event, given that another event will occur.

Definition 2.4.4 [202] *Let $E, F \in \mathcal{E}$ be events with $\Pr[F] > 0$. The conditional probability that E occurs, given that F occurs is*

$$\Pr[E \mid F] = \frac{\Pr[E \cap F]}{\Pr[F]}.$$

From this definition, we have that two events E, F are independent if and only if $\Pr[E \mid F] = \Pr[E]$, respectively $\Pr[F \mid E] = \Pr[F]$. Furthermore, $\Pr[\overline{E} \mid F] = 1 - \Pr[E \mid F]$. The following bound, which we developed specifically for our randomized algorithms, combines the union bound with conditional probabilities.

Proposition 2.4.5 *Let $E_1, \dots, E_n, F \in \mathcal{E}$ be events, with $\Pr[F] > 0$. It holds that*

$$\Pr\left[F \cap \bigcap_{i=1}^{n} E_i\right] \geq 1 - \left(\Pr[\overline{F}] + \sum_{i=1}^{n} \Pr[\overline{E}_i \mid F]\right).$$

Proof. We have:

$$\begin{aligned}
\Pr\left[F \cap \bigcap_{i=1}^{n} E_i\right] &= 1 - \Pr\left[\overline{F} \cup \bigcup_{i=1}^{n} \overline{E}_i\right] \\
&= 1 - \Pr\left[\bigcup_{i=1}^{n} (\overline{E}_i \cap \overline{F}) \cup \bigcup_{i=1}^{n} (E_i \cap \overline{F}) \cup \bigcup_{i=1}^{n} (\overline{E}_i \cap F)\right] \\
&= 1 - \Pr\left[\overline{F} \cup \bigcup_{i=1}^{n} (\overline{E}_i \cap F)\right] \geq 1 - \left(\Pr[\overline{F}] + \sum_{i=1}^{n} \Pr[\overline{E}_i \cap F]\right) \\
&\geq 1 - \left(\Pr[\overline{F}] + \sum_{i=1}^{n} \Pr[\overline{E}_i \mid F]\right).
\end{aligned}$$

In the first equation we use De Morgan's law, the first inequality follows from Proposition 2.4.3, and the last inequality follows, since $\Pr[\overline{E}_i \mid F] \geq \Pr[\overline{E}_i \cap F]$ for all $i \in [n]$. □

2.4.1 Sampling

Sampling is the process of repeatedly drawing elements from some non-empty set with a certain probability, cf. [202]. The result of a sampling process is a multiset S, which we call **sample**. The most popular type of sampling is uniform sampling: we sample **uniformly and independently** (with replacement) from a non-empty and finite set X, if each element of X is drawn with probability $\frac{1}{|X|}$ and each draw is independent of the draws that already took place. Formally, the underlying sample space is $X^{|S|}$, where we neglect the order of the elements of the elementary events. We can easily verify that for each $s \in S$ and $x \in X$ we have $\Pr[s = x] = \frac{1}{|X|}$, by a counting argument. Without loss of generality, we assume that s corresponds to the first draw. There are $1 \cdot |X|^{|S|-1}$ tuples, such that $s = x$, and there is a total of $|X|^{|S|}$ tuples in the sample space, each of which occur with the same probability. Thus, $\Pr[s = x] = \frac{|X|^{|S|-1}}{|X|^{|S|}} = \frac{1}{|X|}$. We now turn to a more involved type of sampling.

Superset Sampling

This technique was coined by Kumar et al. [173] and also applied by Ackermann et al. [6]. Here, we draw a sample S uniformly and independently with replacement from a non-empty finite set X, and we want our sample to contain a uniform and independent sample from a subset $Y \subseteq X$.

Proposition 2.4.6 *Let X be a non-empty finite set and $Y \subseteq X$ be a non-empty subset. Let S be sampled uniformly and independently with replacement from X and let F be the event, that there is a subset $S' \subseteq S$ of size at least $n \in \mathbb{N}$, with $S' \subseteq Y$. For each $s' \in S'$ and $y \in Y$ it holds that $\Pr[s' = y \mid F] = \frac{1}{|Y|}$.*

Proof. The sample space is $X^{|S|}$ and consists of $|X|^{|S|}$ tuples, each of which occurs with the same probability. In $\binom{|S|}{n} \cdot |Y|^n \cdot |X|^{|S|-n}$ of them are at least n elements from Y. Thus, $\Pr[F] = \frac{\binom{|S|}{n} \cdot |Y|^n \cdot |X|^{|S|-n}}{|X|^{|S|}}$.

Without loss of generality, we assume that s' corresponds to the first draw from Y. There are $\binom{|S|}{n} \cdot 1 \cdot |Y|^{n-1} \cdot |X|^{|S|-n}$ tuples such that $s' = y$ and at least n elements are from Y. Hence, $\Pr[(s' = y) \cap F] = \frac{\binom{|S|}{n} \cdot 1 \cdot |Y|^{n-1} \cdot |X|^{|S|-n}}{|X|^{|S|}}$. Finally, we have

$$\Pr[s' = y \mid F] = \frac{\Pr[(s' = y) \cap F]}{\Pr[F]} = \frac{\frac{\binom{|S|}{n} \cdot 1 \cdot |Y|^{n-1} \cdot |X|^{|S|-n}}{|X|^{|S|}}}{\frac{\binom{|S|}{n} \cdot |Y|^n \cdot |X|^{|S|-n}}{|X|^{|S|}}} = \frac{1}{|Y|}.$$

□

2.4.2 Random Variables

Random variables and the associated concepts are central in probability theory. These are functions that depend on the outcome of an underlying probabilistic experiment. In computer science they are often used to analyze some value computed by a randomized algorithm.

Definition 2.4.7 [237, 202] *A (real)* ***random variable*** *is a (measurable) function $X \colon \Xi \to \mathbb{R}$.*

If X takes only a finite or countably infinite number of values, for example when Ξ is finite or countably infinite, then we call X a **discrete random variable**. In the following we write $X = x$ shorthand for the set of events $E \in \Xi$ with $X(E) = x$. The notion of independence naturally extends to random variables.

Definition 2.4.8 [202] *Let $X_1, \ldots, X_n$ be random variables. These are (mutually)* ***independent*** *if and only if for any $I \subseteq [n]$ and $x \in \mathbb{R}$ we have*

$$\Pr\left[\bigcap_{i \in I} (X_i = x)\right] = \prod_{i \in I} \Pr[X_i = x].$$

We can get some information about a random variable by taking its weighted average value, where the weights are with respect to the probability function.

Definition 2.4.9 [237, 202] *Let X be a random variable. The* ***expected value*** *of X is*

$$\mathrm{Exp}[X] = \int_{\Xi} X(E) \; \mathrm{d}\Pr[E].$$

Of course, if X is a discrete random variable, we have

$$\mathrm{Exp}[X] = \sum_{x \in \{X(E) | E \in \Xi\}} x \cdot \Pr[X = x].$$

Or if Ξ is even finite, we have

$$\mathrm{Exp}[X] = \sum_{E \in \Xi} X(E) \cdot \Pr[E].$$

The expected value also extends naturally to conditional probability and since we only use it with respect to discrete variables, we give a constrained definition.

Definition 2.4.10 [202] *Let $E \in \mathcal{E}$ be an event with $\Pr[E] > 0$ and X be a discrete random variable. The* ***conditional expectation*** *of X with respect to E is*

$$\mathrm{Exp}[X \mid E] = \sum_{x \in \{X(F) | F \in \Xi\}} x \cdot \Pr[X = x \mid E].$$

Of course, if Ξ is finite, we have

$$\mathrm{Exp}[X \mid E] = \sum_{F \in \Xi} X(F) \cdot \Pr[F \mid E].$$

The (conditional) expected value is also called **mean** or **expectation**. Furthermore, it has a variety of valuable properties, one of which is the **linearity of expectation**:

Theorem 2.4.11 [237] *Let X, Y be random variables and $a, b \in \mathbb{R}$. It holds that*

$$\mathrm{Exp}[a \cdot X + b \cdot Y] = a \cdot \mathrm{Exp}[X] + b \cdot \mathrm{Exp}[Y]\,.$$

The expected deviation of a random variable from its mean is also often of interest.

Definition 2.4.12 [237, 202] *The* ***variance*** *of a random variable X is defined*

$$\mathrm{Var}[X] = \mathrm{Exp}\Big[(X - \mathrm{Exp}[X])^2\Big]\,.$$

The ***standard deviation*** *of X is defined*

$$\mathrm{std}[X] = \sqrt{\mathrm{Var}[X]}.$$

Probability Distributions

The distribution of probabilities among all subsets of the range of a random variable that correspond to possible events is specified by its **probability distribution**. Informally, by using the probability distribution of a random variable we can abstract from and forget about the underlying probability space. We start by introducing some basic and well-known **discrete probability distributions**.

Definition 2.4.13 [237] *A random variable X follows the* ***discrete uniform distribution*** *if X takes only values $x_1, \dots, x_n \in \mathbb{R}$ and for each $i \in [n]$ we have* $\Pr[X = x_i] = \frac{1}{n}$.

Any random variable defined on an element of a uniform sample from a non-empty finite set naturally follows the discrete uniform distribution. We define another distribution that is related to sampling.

Definition 2.4.14 [202] ***Poisson trials*** *are sequences $X_1, \dots, X_n$ of random variables that only take the values 0 and 1. If $\Pr[X_i = 1] = \Pr[X_j = 1]$ for all $i, j \in [n]$, we call them* ***Bernoulli trials****.*

These types of random variables are often used to model a repeated random experiment that is either successful or unsuccessful.

Contrary to the former, the following distribution is a **continuous probability distribution**. It is extensively used and of central importance in probability theory.

Definition 2.4.15 [237, 202] *A random variable X follows the* ***normal distribution*** *with mean $\mu \in \mathbb{R}$ and standard deviation $\sigma \in (0, \infty)$, if the underlying probability space is $(\mathbb{R}, \mathcal{B}(\mathbb{R}), \Pr)$, where $\mathcal{B}(\mathbb{R})$ denotes the Borel σ-algebra*[1] *of $\mathbb{R}$ and $\Pr\colon \mathcal{B}(\mathbb{R}) \to [0,1]$ is a probability function satisfying*

$$\Pr[x \le X \le y] = \frac{1}{\sqrt{2\pi}\sigma} \int_x^y \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) \, \mathrm{d}z$$

for all $x, y \in \mathbb{R} \cup \{-\infty, \infty\}$ with $x \le y$.

Note that by definition we have $\Pr[X = x] = 0$ for all $x \in \mathbb{R}$, thus

$$\Pr[x < X < y] = \Pr[x \le X < y] = \Pr[x < X \le y] = \Pr[x \le X \le y].$$

The sum of two independent normally distributed random variables with means μ_1 and μ_2 and variances σ_1^2 and σ_2^2 is again normally distributed with mean $\mu_1 + \mu_2$ and variance $\sigma_1^2 + \sigma_2^2$ [237]. Furthermore, the product of a constant $a \in \mathbb{R}$ with a random variable that follows the **standard normal distribution**, i.e., the normal distribution with zero mean and unit variance, is a normally distributed random variable with zero mean and variance a^2 [237].

Finally, the sum of k independent squared random variables that follow the standard normal distribution is a random variable that follows the **Chi-squared distribution** with k degrees of freedom [237] and has mean k [159].

[1] We omit the formal definition and only provide this information for the sake of completeness.

2.4.3 Concentration Inequalities

Concentration inequalities provide upper bounds on the probability that a random variable deviates from its mean. In the context of randomized algorithms, they are often useful for proving that certain bad events are unlikely to happen. Arguably the most basic among them is **Markov's inequality**:

Theorem 2.4.16 [202] *Let X be a random variable with $X(E) \geq 0$ for each $E \in \Xi$. Then for all $x \in \mathbb{R}_{>0}$ it holds that*

$$\Pr[X \geq x] \leq \frac{\operatorname{Exp}[X]}{x}.$$

More advanced and distribution-dependent inequalities can be obtained on top of Markov's inequality by applying the Chernoff method, cf. [41]. These inequalities are named **Chernoff bounds**.

Theorem 2.4.17 [202] *Let $X_1, \ldots, X_n$ be independent Poisson trials. For $\delta \in (0,1)$ it holds that*

$$\Pr\left[\sum_{i=1}^{n} X_i \leq (1-\delta)\operatorname{Exp}\left[\sum_{i=1}^{n} X_i\right]\right] \leq \exp\left(-\frac{\delta^2}{2}\operatorname{Exp}\left[\sum_{i=1}^{n} X_i\right]\right).$$

Theorem 2.4.18 [41] *Let X be a random variable that follows the Chi-squared distribution with k degrees of freedom. For any $a \in (0,\infty)$ it holds that*

$$\Pr[|X-k| \geq 2(\sqrt{ka}+a)] \leq 2\exp(-a).$$

The following theorem combines sampling from a metric space with a Chernoff bound and comes in very handy when one wants to approximate the median of the sample space.

Theorem 2.4.19 [150] *Let $\mathcal{X} = (X, \rho)$ be a metric space and let $\varepsilon \in (0,1]$. Let $T \subseteq X$ and let W be a non-empty sample, drawn uniformly and independently at random from T with replacement. For any fixed $\sigma_1, \sigma_2 \in T$ with $\sum_{\tau\in T}\rho(\tau,\sigma_1) > (1+\varepsilon)\sum_{\tau\in T}\rho(\tau,\sigma_2)$ it holds that*

$$\Pr\left[\sum_{\tau\in W}\rho(\tau,\sigma_1) \leq \sum_{\tau\in W}\rho(\tau,\sigma_2)\right] < \exp\left(-\frac{\varepsilon^2|W|}{64}\right).$$

2.5 Model of Computation and Complexity

Models of computation are the basics to design algorithms and to analyze their complexity. In the following, we define the models relevant to this work.

Definition 2.5.1 [77] *A **random access machine**, in short **RAM**, is a model of computation consisting of a CPU (central processing unit) that runs a program (a finite sequence of instructions indexed by the integers $0, 1, \ldots$) that operates on an infinite array (indexed by the integers $0, 1, \ldots$) of registers – the elementary storage cells, which can store arbitrary integer values. The CPU maintains a program counter that determines the instruction to execute. It is initially set to 0 and after a instruction is executed (except for a branch instruction) it is incremented by one. The set of instructions is restricted and mainly consists of addition, subtraction and memory transfer with direct and indirect addressing. The model does not support a distinguished input to the program, rather the input is encoded into the program. Initially, all registers are* *set to zero and when the program halts – when all instructions have been executed or a negative address is encountered in indirect addressing –, the output is the content of the registers at halting time. The running time of the instructions, except for those loading the input (which take one time step per value to load) is assumed to be either one (unit cost model) or proportional to the number of bits needed to encode the values (logarithmic cost model). The latter is more realistic since any machine that can only store integers from a bounded range needs a logarithmic number of registers to store a large integer.*

A great benefit of the RAM (under the logarithmic cost model) is that it can be simulated by a Turing machine with only polynomial running time blowup [77, 17, 225]. Therefore, for a given problem we can show membership of classic complexity classes, such as P and NP, by defining and analyzing a RAM. However, this comparatively old model is not adequate for modern computers. The instruction set is too restricted, furthermore it is not viable not to distinguish between program and input and to assume that that arbitrary integers can be stored in the registers (when not using the logarithmic cost model). The following model addresses these issues.

Definition 2.5.2 [169, 115, 126] *A **word RAM** a modification of the RAM model. Except for the following, it does not differ from the RAM model. First, it is parameterized by a fixed number $w \in \mathbb{N}$ and the registers can store so-called words – integers in the range $0, \ldots, 2^w - 1$, represented by strings of w bits. Since real computers can only store values consisting of a bounded number of bits in their elementary storage cells, this is more realistic than assuming that the registers can store arbitrary integers. Second, it has a rich instruction set, consisting among others of conditional and unconditional jumps for program flow, integer modular arithmetic, integer division and remainder, bit-shifting and bitwise boolean operations. This is in line with the basic instructions that modern CPUs possess. Each instruction is assumed to have constant running time and the model has a distinguished input to the program, which initially is suitably encoded into the registers.*

To ensure that the word RAM can store pointers to the given data elements, it is usually assumed [116, 117] that $w \geq \log n$, where n is the number of input elements. This is the so-called **transdichotomous** assumption and it is very reasonable, since one wants to analyze the running time depending on the number of input elements and independent of their size.

The word RAM can also be simulated with polynomial running time blowup by a Turing machine, when w is at most polynomial in n [204, Proposition 1.4 and above]. It is nowadays the canonical model used in large parts of computer science [97].

However, in some fields, like computational geometry, it is generally crucial to compute analytic functions (like roots and trigonometric functions) exactly, for example for deciding if two geometric objects intersect. In these computations, irrational numbers naturally arise, which can neither be stored in the word RAM model nor in the classic RAM model. To circumvent these technical difficulties, the following model was introduced, which is the canonical model in computational geometry. If not mentioned otherwise, this is the **standard model** used in this work.

Definition 2.5.3 [235, 219] *A **real RAM** is another modification of the RAM model, developed specifically for use in computational geometry. It does not differ from the RAM model, except that it has a distinguished input (like the word RAM), that the registers can store arbitrary real numbers and that the available instructions include exact arithmetic and analytic functions such as trigonometric functions, the exponential function and logarithms.*

In this work, we also allow use of rounding functions $\lceil\cdot\rceil$ and $\lfloor\cdot\rfloor$ in unit time, whenever the resulting integers consists of $O(\log n)$ bits.[2]

A major drawback of the real RAM is that it can not be simulated by a Turing machine in general, cf. [45], due to the problem of representing real numbers. This shows that it is a highly unrealistic (in the practical sense) assumption that arbitrary real numbers can be stored in the registers. However, “without this assumption it is virtually impossible to prove the correctness of any geometric algorithms” [194], cf. [168]. Nevertheless, through the problems of simulating arbitrary real RAMs on classic Turing machines, it is in general not an easy task to prove membership of a classic complexity class for a geometric problem, cf. [196]. While this is possible for some (decision) problems (e.g. by assuming that the input is rational and avoiding roots), it depends heavily on the problem at hand and in fact, there exists an own branch of complexity theory for computation with real numbers, cf. [38, 37].

Fortunately, recent research has shown that there is no necessity on using real numbers for most geometric algorithms. Rather, one can rely on sufficiently precise approximations by rational numbers, which requires only a bounded number of bits, cf. [228]. The number of bits needed to correctly run a geometric algorithm is called its **input-precision** and is input dependent [97]. In the course, the following model was developed.

Definition 2.5.4 [97] *A **novel real RAM** (introduced under the name real RAM – we call it novel real RAM to distinguish from the “old” real RAM) is an extension of the word RAM (under the transdichotomous assumption). This model combines the realistic properties of the word RAM with the simplifying properties of the real RAM. It has two types of registers, **word registers**, for storing words, and **real registers**, for storing reals. In contrast to the other models it has a limited number of registers, namely 2^w of each type and for computations on the word registers it has the same instructions available as the word RAM. However, computations on the real registers can only comprise arithmetic and square roots. To prevent unnatural computing power it can only cast words to reals by memory transfer from the word registers to the real registers. The other direction is not allowed. Similar to the word RAM its input is initially suitably encoded into the real and word registers. For details, see [97].*

Before we define the relevant complexity classes, we introduce a general **randomness assumption** on all models, i.e., we assume that all models have access to an infinite string of random bits whose values are determined by fair and independent coin tosses. We assume that $O(\log n)$ consecutive unread random bits can be read in unit time, where n is the number of input elements. Therefore, uniform sampling from $[n]$ can be carried out in time $O(1)$.

2.5.1 Complexity Classes

We start with the fundamental complexity classes from classic complexity theory.

Definition 2.5.5 [23] *By* P *we denote the class of decision problems that can be decided by an algorithm with polynomial running time on a RAM (logarithmic cost model[3]).*

Definition 2.5.6 [23] *By* NP *we denote the class of decision problems where a solution (also called certificate or witness) to an instance of the problem can be verified together with the instance by an algorithm with polynomial running time on a RAM (logarithmic cost model).*

[2] This restriction is necessary since otherwise PSPACE-complete problems can be solved in polynomial time [232].

[3] This is only necessary when numbers occur in the computation that are super-polynomial in the input size.

Definition 2.5.7 [23] *By* PSPACE *we denote the class of problems that can be decided by an algorithm with polynomial space on a RAM (logarithmic cost model).*

The following complexity class has recently been defined in the context of geometric algorithms and algorithmic game theory, cf. [231].

Definition 2.5.8 [231, 97] *By* $\exists\mathbb{R}$ *we denote the class of decision problems that can be reduced to the problem of deciding whether an ETR formula is true or false with polynomial running time on a novel real RAM. An ETR (existential theory of the reals) formula is an existentially quantified formula over any polynomials (involving variables and the constants* 0 *and* 1*) compared by* $<, \leq, =, \geq, >$ *and connected by* $\wedge, \vee, \neg$ *and* $\iff$ *.*

This class fits into the landscape of classic complexity theory [23, 231]:

$$\mathrm{P} \subseteq \mathrm{NP} \subseteq \exists\mathbb{R} \subseteq \mathrm{PSPACE}\,.$$

To show that a problem is contained in $\exists\mathbb{R}$, if possible, one can phrase it as a discrete decision problem and design a real verification algorithm that decides the problem.

Definition 2.5.9 [97] *A* ***discrete decision problem*** Q *is a function from the set of all finite integer sequences to the set of truth values* $\{0,1\}$*. A* ***real verification algorithm*** A *for* Q *is an algorithm such that, for a constant* $c \geq 1$ *and any instance* I *to the problem,*

- *A has running time at most n^c on a novel real RAM with parameter $w \leq c \cdot \log n$, where n is the number of input values,*
- *if $Q(I) = 1$ there exists a solution for I, consisting of a sequence of integer values and a sequence of real values, both of size at most $|I|^c$, which A accepts when given as input together with I and*
- *if $Q(I) = 0$ then A rejects any two sequences of real values and integer values that are given together with I as input.*

The following powerful theorem enables the aforementioned insight.

Theorem 2.5.10 [97] *Any discrete decision problem that has a real verification algorithm is contained in* $\exists\mathbb{R}$.

2.6 Range Spaces

Range spaces are structures, which are mainly studied in the context of sampling in the field of statistical learning theory, cf. [245]. We start with a formal definition.

Definition 2.6.1 *A* ***range space*** *is a pair* $(X, \mathcal{R})$*, where* X *is a set, called ground set and* $\mathcal{R}$ *is a set of subsets* $R \subseteq X$*, which are called ranges.*

Range spaces can be projected onto a subset of the ground set.

Definition 2.6.2 [130] *The* ***projection*** *of* $(X, \mathcal{R})$ *onto a subset* $Y \subseteq X$ *is the range space* $(Y, \mathcal{R}_{|Y})$*, where we denote* $\mathcal{R}_{|Y} = \{R \cap Y \mid R \in \mathcal{R}\}$.

Further, each range space has a complementary range space.

Definition 2.6.3 [130] *The **complementary range space** of $(X, \mathcal{R})$ is $(X, \overline{\mathcal{R}})$, where $\overline{\mathcal{R}} = \{X \setminus R \mid R \in \mathcal{R}\}$.*

We say that a subset $Y \subseteq X$ is shattered by $\mathcal{R}$, if $\mathcal{R}_{|Y}$ contains all subsets of Y. A measure of the combinatorial complexity of a range space is the VC dimension.

Definition 2.6.4 [130] *The **VC (Vapnik-Chervonenkis) dimension** of $(X, \mathcal{R})$ is the maximum cardinality of a shattered subset of X.*

Note that F and $\overline{F}$ have equal VC dimension and for any $Y \subseteq X$, the projection of F onto Y has VC dimension at most the VC dimension of F, see for example [130].

Range spaces do not have to be finite and can be discretized by means of ε-nets.

Definition 2.6.5 [130] *A set $N \subseteq X$ is an ε-**net** for $(X, \mathcal{R})$ if for any range $R \in \mathcal{R}$, we have $R \cap N \neq \emptyset$ when $|R \cap X| \geq \varepsilon |X|$.*

A subsystem oracle can be used to compute ε-nets deterministically.

Definition 2.6.6 [53] *Let $(X, \mathcal{R})$ be a finite range space. A **subsystem oracle** is an algorithm which for any $Y \subseteq X$, lists all sets in $\mathcal{R}_{|Y}$ in time $O(|Y|^{\mathcal{D}+1})$, where $\mathcal{D}$ is the VC dimension of $(X, \mathcal{R})$.*

We use the following theorem to obtain ε-nets when provided with a subsystem oracle.

Theorem 2.6.7 [53] *Let $(X, \mathcal{R})$ be a range space with finite ground set and VC dimension $\mathcal{D}$, and $\varepsilon > 0$ be a given parameter. Assume that there is a subsystem oracle for $(X, \mathcal{R})$. Then an ε-net of size $O\left(\frac{\mathcal{D}}{\varepsilon} \log \frac{\mathcal{D}}{\varepsilon}\right)$ can be computed deterministically in time $O\left(\mathcal{D}^{3\mathcal{D}} \cdot \left(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon}\right)^{\mathcal{D}} \cdot |X|\right)$.*

We define the (ε, η)-approximation of a range space, which also yield a form of discretization that captures the properties of the range space.

Definition 2.6.8 [136] *Let $\varepsilon, \eta \in (0, 1)$ and $(X, \mathcal{R})$ be a range space with finite non-empty ground set. An (η, ε)-**approximation** of $(X, \mathcal{R})$ is a set $S \subseteq X$, such that for all $R \in \mathcal{R}$*

$$\left| \frac{|R \cap X|}{|X|} - \frac{|R \cap S|}{|S|} \right| \leq \begin{cases} \varepsilon \cdot \frac{|R \cap X|}{|X|}, & \text{if } |R \cap X| \geq \eta \cdot |X| \\ \varepsilon \cdot \eta, & \text{else.} \end{cases}$$

The following theorem is useful for obtaining (ε, η)-approximations. In particular, the beneficial link between VC dimension and sampling can be observed.

Theorem 2.6.9 [136] *Let $(X, \mathcal{R})$ be a range space with finite non-empty ground set and VC dimension $\mathcal{D}$. Also, let $\varepsilon, \delta, \eta \in (0, 1)$. There is an absolute constant $c \in \mathbb{R}_{>0}$ such that a sample of*

$$\frac{c}{\eta \cdot \varepsilon^2} \cdot \left(\mathcal{D} \log \left(\frac{1}{\eta} \right) + \log \left(\frac{1}{\delta} \right) \right)$$

elements drawn independently and uniformly at random with replacement from X is a (η, ε)-approximation for $(X, \mathcal{R})$ with probability at least $1 - \delta$.

2.7 Computing Distance Functions

Computing distance functions between spatial data sequences is usually quite costly in terms of running time. For example the discrete Fréchet distance between two point sequences σ and τ can be computed in time $O(|\sigma| \cdot |\tau|)$ by a dynamic program, which we omit here, since the discrete Fréchet distance is not in the focus of this work. The p-dynamic time warping distance can be computed by a similar dynamic program in equal time.

2.7.1 Dynamic Time Warping Distance

The canonical algorithm for computing the dynamic time warping distance between two point sequences is by Sakoe and Chiba [227]. To compute the p-dynamic time warping distance we slightly modify their dynamic program to handle values of p other than one:

Algorithm 1 p-Dynamic Time Warping Distance

```
procedure $p$-DTW$(\sigma = (\sigma_1, \dots, \sigma_{|\sigma|}), \tau = (\tau_1, \dots, \tau_{|\tau|}))$
    $D \leftarrow \text{array}[0 \dots |\sigma|][0 \dots |\tau|]$
    for $i = 0, \dots, |\sigma|$ do
        for $j = 0, \dots, |\tau|$ do
            $D[i][j] \leftarrow \infty$
    $D[0][0] \leftarrow 0$
    for $i = 1, \dots, |\sigma|$ do
        for $j = 1, \dots, |\tau|$ do
            $D[i][j] \leftarrow \vartheta(\sigma_i, \tau_j)^p + \min\{D[i-1][j], D[i][j-1], D[i-1][j-1]\}$
    return $\sqrt[p]{D[|\sigma|][|\tau|]}$
```

Since this algorithm is a straight-forward adaption, the following corollary follows from the result of Sakoe and Chiba [227].

Corollary 2.7.1 *Given two point sequences σ and τ, Algorithm 1 computes the p-dynamic time warping distance in time*[4] *$O(|\sigma| \cdot |\tau|)$.*

We note that recently a slightly faster algorithm for computing the dynamic time warping distance for the special case that ϑ is a distance induced by a norm whose unit ball is a symmetric polytope with a constant number of facets, each of constant complexity, was developed. For example, this is the case when $\vartheta(p,q) = \|p-q\|_1$ or $\vartheta(p,q) = \|p-q\|_\infty$, where $\|\cdot\|_1$, respectively $\|\cdot\|_\infty$, denotes the ℓ_1 norm, respectively ℓ_∞ norm. The algorithm has running time $O(n^2 \log\log\log n / \log\log n)$ for two sequences of complexity n each [122].
On the other hand, for any $\delta > 0$ there is no $O(n^{2-\delta})$ time algorithm for computing the dynamic time warping distance, unless the Strong Exponential Time Hypothesis fails [52]. The Strong Exponential Time Hypothesis states that for all $\delta > 0$ there is some $k \geq 3$ such that there is no $O(2^{(1-\delta)n})$ time algorithm that solves k-SAT (the satisfiability problem for boolean formulas of at most k literals in a clause) for formulas of n variables. Though, there exist a near-linear time $(1+\varepsilon)$-approximation algorithm for a restricted class of point sequences in $\mathbb{R}^d$ [12].

[4] For simplicity, we assume that ϑ can be evaluated in constant time.

2.7.2 Fréchet Distance

The canonical algorithm for computing the Fréchet distance between two polygonal curves is by Alt and Godau [21]. This algorithm is based on repeatedly solving the decision problem whether $\mathrm{d_F}(\sigma, \tau) \leq r$ for a given $r \in [0, \infty)$. To solve this decision problem the algorithm uses the concept of free space, which we now introduce.

Definition 2.7.2 [21] *The r-**free space**, where $r \in [0, \infty)$, between two line segments $s_1 = \overline{p_1 p_2}$ and $s_2 = \overline{q_1 q_2}$ (where $p_1, p_2, q_1, q_2 \in \mathbb{R}^d$) is the set*

$$\{(\lambda_1, \lambda_2) \in [0,1]^2 \mid \|\mathrm{lp}\,(s_1, \lambda_1) - \mathrm{lp}\,(s_2, \lambda_2)\| \leq r\}.$$

*The r-**free space** between two polygonal curves $\sigma, \tau \in \mathbb{R}^d_*$ is the set*

$$\bigcup_{\substack{i \in [|\sigma|-1] \\ j \in [|\tau|-1]}} \{(\lambda_1 + i, \lambda_2 + j) \mid (\lambda_1, \lambda_2) \in F_{i,j}\},$$

where $F_{i,j}$ is the r-free space between the i^{th} edge of σ and the j^{th} edge of τ.

The r-free space between two line segments is called a **free space cell** and is a convex set. The whole r-free space is called the **free space diagram**.

Proposition 2.7.3 [21] *For any $r \in [0, \infty)$ and any two line segments the r-free space is a convex set.*

Alt and Godau's central insight is that $\mathrm{d_F}(\sigma, \tau) \leq r$ if and only if there exists a curve γ in the r-free space with $\gamma(0) = (1, 1)$ and $\gamma(1) = (|\sigma|, |\tau|)$ and whose coordinates are non-decreasing. This curve corresponds to the limit of a sequence of matchings realizing the distance r between the curves, see Proposition 2.3.12.

Proposition 2.7.4 [21] *For any $r \in [0, \infty)$ and any two polygonal curves $\sigma, \tau \in \mathbb{R}^d_*$ we have $\mathrm{d_F}(\sigma, \tau) \leq r$ if, and only if, there exists a curve γ in the r-free space between σ and τ, with $\gamma(0) = (1, 1)$, $\gamma(1) = (|\sigma|, |\tau|)$ and that is monotone in both coordinates.*

Since the cells of the r-free space are convex, the existence of such a curve can be checked by computing only the borders of the cells, which can be done efficiently.

Theorem 2.7.5 [21] *There exists an algorithm that, given an $r \in [0, \infty)$ and two polygonal curves $\sigma, \tau \in \mathbb{R}^d_*$, decides whether $\mathrm{d_F}(\sigma, \tau) \leq r$ and has running time $O(|\sigma| \cdot |\tau|)$.*

Finally, Alt and Godau have shown that for every two polynomial curves σ, τ there exists only a finite number (depending only on the complexities of the curves) of values – the so-called *critical values* –, of which one is their Fréchet distance. These values are determined by

(1) the distances between the first and last vertices of the curves (which must be matched to each other),

(2) distances between vertices and edges and

(3) the common distance of two vertices on one curve to the intersection of their bisector with an edge of the other curve.

We call (1) a vertex event, (2) an edge event and (3) a monotonicity event. By parametric search on the sorted sequence of the critical values one can compute the Fréchet distance. The parametric search is guided by solving the decision problem $\mathrm{d_F}(\sigma, \tau) \leq r$.

Theorem 2.7.6 [21] *There exists an algorithm that, given two polygonal curves $\sigma, \tau \in \mathbb{R}^d_*$, computes $\mathrm{d_F}(\sigma, \tau)$ and has running time $O(|\sigma| \cdot |\tau| \cdot \log(|\sigma| \cdot |\tau|))$.*

We note that for any $\delta > 0$ there is neither a $O(n^{2-\delta})$ time algorithm for computing the discrete nor the continuous Fréchet distance, unless the Strong Exponential Time Hypothesis[5] fails [51]. Though, there exists an algorithm that runs in expected time $O(n^2\sqrt{\log n}(\log\log n)^{3/2})$ for two curves of complexity n each [57, 58] and a $(1 + \varepsilon)$-approximation algorithm for a restricted class of polygonal curves that runs in near-linear time [85, 86].

2.8 Simplification

Simplification is a problem that is strongly related to compression. It appears in many problems concerning spatial data sequences as a sub-problem, often to deal with overfitting or to reduce running time. Here, we are given a point sequence or polygonal curve and we want to compute another point sequence/curve that has small distance to the original sequence/curve and is of smaller complexity, where the target complexity is a parameter of the problem. The idea is that the low-complexity sequence/curve can be used as a substitute for the high-complexity sequence/curve, since it is geometrically close to it by the (weak) triangle inequality.

2.8.1 Point Sequences

Here we focus on simplifications of point sequences with respect to the p-dynamic time warping distance.

Definition 2.8.1 *For a point sequence $\tau \in M^m$ we denote by $\mathrm{simpl}(\ell, \alpha, \tau)$ an α-approximate minimum-error ℓ-simplification of τ, i.e., a point sequence $\sigma \in M^{\leq \ell}$ with $\mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma) \leq \alpha \cdot \mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma')$ for all $\sigma' \in M^{\leq \ell}$.*

We present a dynamic programming solution for the problem of computing such an approximate minimum-error simplification. Our algorithm can be seen as a special case of the result of Brill et al. [49, 50] for computing a mean of restricted complexity, but since our statement is different, we include a proof for completeness.

[5] Actually, the author even proves that this is the case unless a weaker variant of the Strong Exponential Time Hypothesis fails, but this implies the stated result.

Algorithm 2 2-Approximate Minimum Error Simplification under p-DTW

```
procedure DTW-Simplification($\tau = (\tau_1, \dots, \tau_m), \ell, p$)
    $C \leftarrow$ array$[1 \dots m][1 \dots \ell]$ with elements in $M^{\leq \ell}$
    $D \leftarrow$ array$[1 \dots m][1 \dots \ell]$ with elements in $\mathbb{R}_{\geq 0}$
    $P \leftarrow \{\tau_1, \dots, \tau_m\}$
    for each $i = 1, \dots, m$ do
        for each $j = 1, \dots, \ell$ do
            if $j = 1$ then
                $q^* \leftarrow$ arbitrary element from $\arg\min_{q \in P} \sum_{k=1}^{i} \vartheta(\tau_k, q)^p$
                $D[i][j] \leftarrow \sum_{k=1}^{i} \vartheta(\tau_k, q^*)^p$;  $C[i][j] \leftarrow (q^*)$
            else
                $i' \leftarrow$ arbitrary element from $\arg\min_{k' \in [i]} \left( D[k'][j-1] + \min_{q \in P} \sum_{k=k'}^{i} \vartheta(\tau_k, q)^p \right)$
                $q^* \leftarrow$ arbitrary element from $\arg\min_{q \in P} \sum_{k=k'}^{i} \vartheta(\tau_k, q)^p$
                $D[i][j] \leftarrow D[i'][j-1] + \sum_{k=i'}^{i} \vartheta(\tau_k, q^*)^p$;  $C[i][j] \leftarrow C[i'][j-1] \oplus (q^*)$
    $j^* \leftarrow$ arbitrary element from $\arg\min_{j \in [\ell]} D[m][j]$
    return $C[m][j^*]$
```

We first show that no simplification that can be constructed from τ's vertices is better than the result of Algorithm 2.

Lemma 2.8.2 *Given as input a point sequence $\tau = (\tau_1, \dots, \tau_m) \in M^m$, Algorithm 2 returns a point sequence from $P^{\leq \ell}$ that minimizes the p-DTW distance to τ among all point sequences in $P^{\leq \ell}$, where $P = \{\tau_1, \dots, \tau_m\}$.*

Proof. We show that $C[m][j^*]$ satisfies

$$\mathrm{d}_{\mathrm{DTW}_p}(\tau, C[m][j^*]) = \min_{\sigma' \in P^{\leq \ell}} \mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma').$$

We claim that there is a point sequence $\sigma \in P^{\leq \ell}$ such that

$$\mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma) = \min_{\sigma' \in P^{\leq \ell}} \mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma'),$$

and such that the optimal warping between τ and σ does not match two vertices of σ with the same vertex of τ. To see this, consider an optimal warping $W \in \mathcal{W}_{m,|\sigma'|}$ between τ and some point sequence $\sigma' = (\sigma'_1, \dots, \sigma'_j) \in P^{\leq \ell}$. Let $(i, j) \in W$ and $(i, j+1) \in W$. If $(i-1, j) \in W$ then removing (i, j) yields a new warping with a cost at most equal to the cost of W. Similarly, if $(i+1, j+1) \in W$ then removing $(i, j+1)$ from W yields a new warping with a cost at most equal to the cost of W. If $(i-1, j) \notin W$, then we can remove σ'_j from σ'. If $(i+1, j+1) \notin W$, then we can remove σ'_{j+1} from σ'. We conclude that there exist a point sequence $\sigma'' \in P^{\leq \ell}$ such that $\mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma'') \leq \mathrm{d}_{\mathrm{DTW}_p}(\tau, \sigma')$, and an optimal warping $W \in \mathcal{W}_{m,|\sigma''|}$ between τ and σ'' for which there are no $i \in [m], j \in [\ell]$ such that both $(i, j) \in W$ and $(i, j+1) \in W$.

For each $i \in [m]$, let $\tau_{|i} = (\tau_1, \dots, \tau_i)$. By construction, each $D[i][j]$ stores the minimum distance between $\tau_{|i}$ and any point sequence σ from P^j, where the distance is attained by a warping that does not match two vertices of σ to the same vertex of τ. Hence, $D[m][j^*]$ stores the minimum distance between τ and any point sequence in $P^{\leq \ell}$, and $C[m][j^*]$ stores a point sequence from $P^{\leq \ell}$ with distance $D[m][j^*]$ from τ. □

Now we show that this simplification is a 2-approximate minimum-error ℓ-simplification among all possible minimum-error ℓ-simplifications.

Lemma 2.8.3 *Given as input a point sequence $\tau = (\tau_1, \dots, \tau_m) \in M^m$, Algorithm 2 returns a 2-approximate minimum-error ℓ-simplification under the p-DTW distance.*

Proof. Let $P = \{\tau_1, \dots, \tau_m\}$. By Lemma 2.8.2, $C[m][j^*]$ is a point sequence in $P^{\leq \ell}$ that minimizes the distance to τ among all point sequences in $P^{\leq \ell}$.

We show that $C[m][j^*]$ is a 2-approximate minimum-error ℓ-simplification. Let $\tau^* = (\tau_1^*, \dots, \tau_{|\tau^*|}^*)$ be an optimal (1-approximate) minimum-error ℓ-simplification of τ, and let $\sigma^* = (\sigma_1^*, \dots, \sigma_{|\tau^*|}^*)$, where for each $i \in [|\tau^*|]$ we let $\sigma_i^* \in \arg\min_{q \in P} \vartheta(q, \tau_i^*)$ be arbitrary. Let $W^* \in \mathcal{W}_{m,|\tau^*|}$ be an optimal warping between τ and τ^*. Then,

$$\begin{aligned}
\mathrm{d_{DTW}}_p^1(\tau, C[m][j^*]) \leq \mathrm{d_{DTW}}_p^1(\tau, \sigma^*) &= \min_{W \in \mathcal{W}_{m,|\tau^*|}} \left(\sum_{(i,j) \in W} \vartheta(\tau_i, \sigma_j^*)^p \right)^{1/p} \\
&\leq \left(\sum_{(i,j) \in W^*} \vartheta(\tau_i, \sigma_j^*)^p \right)^{1/p} \leq \left(\sum_{(i,j) \in W^*} \left(\vartheta(\tau_i, \tau_j^*) + \vartheta(\tau_j^*, \sigma_j^*) \right)^p \right)^{1/p} \qquad \text{(I)} \\
&\leq \left(\sum_{(i,j) \in W^*} 2^p \vartheta(\tau_i, \tau_j^*)^p \right)^{1/p} \quad = 2\, \mathrm{d_{DTW}}_p(\tau, \tau^*),
\end{aligned}$$

where in Eq. (I) we applied the triangle inequality. □

We use the above lemmas to prove the correctness and further analyze the running time of Algorithm 2.

Theorem 2.8.4 *Given as input a point sequence $\tau \in M^m$, Algorithm 2 computes a 2-approximate minimum-error ℓ-simplification of τ under the p-DTW distance in time $O(m^4 \ell)$.*

Proof. Correctness of Algorithm 2 follows from Lemma 2.8.3. It remains to bound the running time of the algorithm. To do so, we consider the operations taking place in the body of the nested loop. For each i, j, we iterate over $O(m)$ values for i' and for each value of k' we compute $\min_{x \in P} \sum_{k=k'}^{i} \vartheta(\tau_k, q)^p$ in time $O((i - k') \cdot m) = O(m^2)$. Hence, the total running time is $O(m^4 \ell)$. □

2.8.2 Polygonal Curves

Simplifications for polygonal curves under the Fréchet distance are defined analogously.

Definition 2.8.5 *For a polygonal curve $\tau \in \mathbb{R}^d_*$ we denote by $\mathrm{simpl}(\ell, \alpha, \tau)$ an α-approximate minimum-error ℓ-simplification of τ, i.e., a curve $\sigma \in \mathbb{R}^d_\ell$ with $\mathrm{d_F}(\tau, \sigma) \leq \alpha \cdot \mathrm{d_F}(\tau, \sigma')$ for all $\sigma' \in \mathbb{R}^d_\ell$.*

We can use an existing approach by Imai and Iri [149] that, similar to Algorithm 2, computes a simplification using the vertices of the original curve. When this approach is combined with the algorithm by Alt and Godau [21] it can be used to compute simplifications with respect to the Fréchet distance.

Theorem 2.8.6 [59] *Given a curve $\sigma \in \mathbb{R}^d_m$, a 4-approximate minimum-error ℓ-simplification can be computed in $O(m^3 \log m)$ time.*

Unfortunately, the approximation factor of this approach is twice the approximation factor of the approach for point sequences. This is due to the linear interpolation and also shows in the relation between Fréchet and discrete Fréchet distance:

Let $\tau, \sigma \in \mathbb{R}^d_*$ with vertices $v^\tau_1, \dots, v^\tau_{|\tau|}$, respectively $v^\sigma_1, \dots, v^\sigma_{|\sigma|}$. It holds that [95]

$$\mathrm{d_{dF}}((v^\tau_1, \dots, v^\tau_{|\tau|}), (v^\sigma_1, \dots, v^\sigma_{|\sigma|})) \leq \mathrm{d_F}(\tau, \sigma) + \max \left\{ \max_{i \in [|\tau|-1]} \|v^\tau_i - v^\tau_{i+1}\|, \max_{i \in [|\sigma|-1]} \|v^\sigma_i - v^\sigma_{i+1}\| \right\}$$

and

$$\mathrm{d_F}(\tau, \sigma) \leq \mathrm{d_{dF}}((v^\tau_1, \dots, v^\tau_{|\tau|}), (v^\sigma_1, \dots, v^\sigma_{|\sigma|})).$$

3 Median of Spatial Data Sequences

This chapter is twofold, the first part is dedicated to the problem of computing a median of a set of polygonal curves in the Euclidean space with respect to the Fréchet distance and the second part is dedicated to the problem of computing a median of a set of point sequences in an arbitrary metric space with respect to the dynamic time warping distance. These problems are adaptions of the geometric median of points in the Euclidean space, in the sense that both minimize the sum of distances between the given objects and the median object. The geometric median itself is a generalization of the well-known statistical median [98].

The geometric median problem was probably first considered by Pierre de Fermat in 1643 with a pure geometrical motivation: he sought to find a point that minimizes the sum of distances to three given points in the (Euclidean) plane. Much later in 1909, Alfred Weber studied a facility location problem, i.e., locate a facility in such a way that the sum of costs of transporting goods to the facility is minimized, thereby he obtained a generalized variant of Fermat's problem where the objective is to find a point that minimizes the sum of distances to a given set of n points (in the Euclidean plane), cf. [93]. The nowadays geometric median problem is a simple generalization of Weber's problem to a set of points in $\mathbb{R}^d$.

The geometric median is a statistic that should be of consideration whenever a measure of central tendency of a set of points under the presence of outliers is needed. This is due to its three central properties [93]:

- **Robustness**: It is stable when up to 50% of the points are *arbitrarily* corrupted, e.g. by measurement noise. This means that more than 50% outliers are necessary to move the median outside the range of the non-outliers.
- **Uniqueness**: For each non-empty set of points in $\mathbb{R}^d$ there exists a median and the median is unique whenever the points are not collinear.
- **Equivariance**: The geometric median is equivariant under Euclidean motions, i.e., if we apply a motion to the given point set then its median is the median of the original point set with the same motion applied.

A drawback from the computational side is that the geometric median can generally not be computed exactly for $d \geq 2$ using only arithmetic operations and k^{th} roots [29, 28] – it can also not be constructed by hand using ruler and compass [200]. However, already in 1937 an iterative procedure that produces a sequence of points converging to the geometric median was discovered [250, 251]. Furthermore, modern optimization algorithms are capable of approximating the median to an arbitrary level of accuracy and even run in near-linear time, see e.g. [75].

The appealing aggregation properties (robustness and uniqueness), as well as the computational tractability of the geometric median motivate us to study variants of the problem adapted to spatial data sequences. Aggregation is particularly important in a Big Data context, but also in the general problem of analyzing the underlying phenomenon of the given observations (the data set at hand) and aggregation of spatial data sequences has to this day been studied extensively, but the literature is quite heterogeneous.

One line of research studies the problem of computing a so-called *average sequence*, or (weighted) *Fréchet mean* with respect to the dynamic time warping distance [208, 209, 215, 214, 216, 49, 50, 230]. An average sequence, or Fréchet mean, is a sequence that minimizes the sum of (weighted) squared distances to the given set of sequences. This problem is particularly popular in the data mining and computational biology community with applications ranging from speech and signal recognition to DNA alignment.

Another line of research studies the problem of computing a so-called *middle curve*, or *mean curve* with respect to (variants of) the Fréchet distance [134, 135, 15, 16, 64]. Here, the data sequences are interpreted as polygonal curves and the objective is to compute a curve that minimizes the maximum distance to the input curves. This problem is particularly relevant in the geographic information systems and data mining communities with applications such as handwriting recognition, trajectory analysis and time series analysis in general.

Another very different approach studies the problem of computing a so-called *median trajectory* [55, 56]. Here, a set of trajectories, represented as polygonal curves in $\mathbb{R}^2$ that start and end in the same points s and t, is given and a median trajectory is computed using the arrangement of lines induced by the curves. A basic definition, the so-called *simple median*, is the curve obtained by starting in s and ending in t, while always following the median level line segment in the arrangement. An advanced definition, the so-called *homotopic median* imposes an additional restriction on the median curve. Here, it is assumed that the input curves are homotopic with respect to a certain punctured plane, i.e., there are poles (points) placed in the large faces of the arrangements and all input curves can be continuously deformed one into another without crossing a pole. The homotopic median is obtained similar to the simple median, but we only change the line segment we follow in the arrangement when the resulting curve respects the homotopy type of the input curves. While this approach is certainly interesting and meaningful, there is no obvious extension to polygonal curves in $\mathbb{R}^d$ since those would not induce such an arrangement whenever $d > 2$. This strongly narrows its applicability.

A follow-up approach [244] that was inspired by the aforementioned approach studies the so-called *majority median*, which aims to overcome further shortcomings of the homotopic median: there must not be large faces in the arrangement and further, there must not be a large subset of input trajectories that are homotopic. The majority median is constructed very differently, by using a certain planar graph and graph algorithms, but aims to fulfill similar properties as the simple and the homotopic median.

A related concept [71] follows the direction of using homotopy, using the same setting as in [55, 56]. Here, the objective is to minimize either the maximum homotopy area between the median and the input trajectories or the sum of homotopy areas between the median and the input trajectories. This approach only works if the trajectories do not self-intersect. In this case the homotopy area between two trajectories (polygonal curves) σ, τ is well-defined. The homotopy area of $H \colon [0,1] \times [0,1] \to \mathbb{R}^2$, which is a continuous deformation of σ into τ (this means that $H(t,0) = \sigma(t)$ and $H(t,1) = \tau(t)$ for all $t \in [0,1]$ and H is continuous in both arguments), between σ and τ is $A(\sigma,\tau,H) = \int_0^1 \int_0^1 \left| \frac{H}{\partial t_1} \times \frac{H}{\partial t_2} \right| \mathrm{d}t_1 \; \mathrm{d}t_2$. The homotopy area between σ and τ is $\inf_{H \colon [0,1]\times[0,1]\to\mathbb{R}^2} A(\sigma,\tau,H)$, where the infimum ranges over all continuous deformations of σ into τ. Intuitively, this measure can be seen as a sum-based variation of the Fréchet distance.

However, all of these median trajectory approaches suffer from various limitations. The topic of computing an average sequence or (weighted) Fréchet mean of a set of point sequences under the dynamic time warping distance is still active research and recent lines of research dealing with polygonal curves under the Fréchet distance [87, 59, 60, 207, 44] (and also under the discrete Fréchet [61] and Hausdorff [207] distances) follow and extend variants of the problems of computing an average sequence, respectively a middle or mean curve. These extensions are phrased in the light of k-clustering and are named (k,ℓ)-center clustering and (k,ℓ)-median clustering. Here, one does not only want to compute a curve that aggregates the whole given set of curves, but k curves that each aggregate an element of a k-partition of the set, a hidden structure one wants to uncover. A notable similarity that all these approaches have in common is that one is interested in computing aggregate curves of complexity at most ℓ. If such a bound is not enforced, then the complexity of the aggregate can grow as large as the sum of the complexities of the given curves [181], leading to severe overfitting [98] and overfitting is not desirable since one wants to aggregate the properties of the underlying phenomenon and not the peculiarities of the given curves. Furthermore, the continuous nature of the Fréchet distance actually enables the aggregation of the given curves by a curve of small complexity.

(k,ℓ)-center clustering and (k,ℓ)-median clustering are derived from the famous k-center clustering and k-median clustering methods of points in $\mathbb{R}^d$; we will look at these problems in detail in Chapter 4. For $k=1$ they correspond to the well-known minimum enclosing ball problem [252] and the geometric median problem. Consequently, the $(1,\ell)$-center clustering problem is related to the problem of computing a middle curve as well as computing a median trajectory that minimizes the maximum homotopy area to the input trajectories. The $(1,\ell)$-median clustering problem is related to the problem of computing an average sequence as well as computing a median trajectory that minimizes the sum of homotopy areas to the input trajectories. In this chapter, we study the $(1,\ell)$-median problem for polygonal curves under the Fréchet distance, which we simply call ℓ-median problem, and the related (p,q)-mean problem for point sequences under the p-dynamic time warping distance. We also call the $(1,\ell)$-center problem the ℓ-center problem for simplicity. In Chapter 4 we extend our techniques and study the (k,ℓ)-median clustering problem and the related (k,ℓ,p,q)-mean clustering problem.

We start our journey by studying the ℓ-median for polygonal curves under the Fréchet distance.

3.1 Polygonal Curves

First, we formally define the problem that we study in this section, then we closely review the related work.

3.1.1 Problem Definition

Arguably the most natural way to formulate this problem is as optimization problem of computing a polygonal curve of bounded (constant) complexity that minimizes the Fréchet distances between the given curves and the median curve.

Problem 3.1.1 *The* ***ℓ-median (optimization) problem*** *is defined as follows, where $\ell \in \mathbb{N}_{>1}$ is a fixed (constant) parameter of the problem: given a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, compute a polygonal curve $c \in \mathbb{R}^d_\ell$, such that $\operatorname{cost}(T,c) = \sum_{i=1}^{n} \mathrm{d_F}(\tau_i, c)$ is minimal.*

The following is a corresponding decision problem. Here we have an additional parameter r and the problem asks whether there exists a polygonal curve of bounded complexity such that the sum of distances to the given curves does not exceed r.

Problem 3.1.2 *The* ***ℓ-median decision problem*** *is defined as follows, where $\ell \in \mathbb{N}_{>1}$ is a fixed (constant) parameter of the problem: given a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{Q}^d_m$ of polygonal curves and a number $r \in \mathbb{Q}$, decide whether there exists a curve $c \in \mathbb{R}^d_\ell$, such that $\operatorname{cost}(T, c) \leq r$.*

It is easy to see that Problem 3.1.1 and Problem 3.1.2 are related: if we have an algorithm A that decides Problem 3.1.2, we can use it as a black box in a modified exponential search [34] to approximate Problem 3.1.1 (with restriction $T \subset \mathbb{Q}^d_m$) within any factor $1 + \varepsilon$, in time $O((\log(r^*) + \log(1/\varepsilon)) \cdot T_A(m, n))$, where $r^* = \arg\min_{c \in \mathbb{R}^d_\ell} \operatorname{cost}(T, c)$ and T_A denotes the running time of A. Note that we may not solve the problem exactly, since we require that $r \in \mathbb{Q}$ and r^* may be an element of $\mathbb{R} \setminus \mathbb{Q}$.

In the following sections, we are shorthand referring to the ℓ-median by median.

3.1.2 Related Work

This ℓ-median problem is relatively unexplored. It has only recently been introduced by Driemel et al. [87] in 2016. They developed the first $(1 + \varepsilon)$-approximation algorithm and a constant factor approximation algorithm for this problem, for curves in $\mathbb{R}$. The algorithms both have running time in $O(nm \log m)$ (for constant ε). Very recently, Buchin et al. [61] proved that the problem is NP-hard and W[1]-hard in n.

Contrary to the related **ℓ-center problem**, for polygonal curves under the Fréchet distance, which asks to compute a polygonal curve in $\mathbb{R}^d$ of complexity at most ℓ that minimizes the maximum Fréchet distance between the center curve and the input curves and was also introduced by Driemel et al. [87], there is no result on the hardness of approximation. The ℓ-center problem is also NP-hard [59] and there exists a $(1 + \varepsilon)$-approximation algorithm and a constant factor approximation algorithm for curves in $\mathbb{R}$ [87], which also have running time in $O(nm \log m)$ (for constant ε). Furthermore, there is a constant factor approximation algorithm for the ℓ-center problem for curves in $\mathbb{R}^d$ with running time $O(nm \log m + m^3 \log m)$ by Buchin et al. [59]. They also proved that the problem is NP-hard to approximate[1] within a factor of $(1.5 - \varepsilon)$ for $d = 1$ and $(2.25 - \varepsilon)$ for $d \geq 2$. Also, they proved that it remains NP-hard even if the restriction on the complexity of the center curve is dropped.

We summarize the main algorithmic results on both problems under the Fréchet distance.

Problem	Approx. Fact.	Running Time	Ambient Space	Reference
ℓ-center	$1 + \varepsilon$	$O(nm \log m)$, ε const.	$\mathbb{R}$	[87]
	3	$O(nm \log m + m^3 \log m)$	$\mathbb{R}^2$	[59]
	6	$O(nm \log m + m^3 \log m)$	$\mathbb{R}^d$	
ℓ-median	$1 + \varepsilon$	$O(nm \log m)$, ε const.	$\mathbb{R}$	[87]
	$1 + \varepsilon$ (bi-criteria)	$n \cdot 2^{O(\varepsilon^{-2} + \log m)}$	$\mathbb{R}^d$	Corollary 3.1.26
	$3 + \varepsilon$ (bi-criteria)			Corollary 3.1.22
	34	$O(m^3 \log m + \log n)$		Corollary 3.1.18
	65	$O((n + \log^5 n) m \log m)$	$\mathbb{R}$	[87]

Both problems and variants of them have also been studied under the discrete Fréchet distance. Just as their counterparts under the continuous Fréchet distance, they are NP-hard [59, 61]. The ℓ-median problem under the discrete Fréchet distance is W[1]-hard in n and the ℓ-center problem

[1] Where ℓ is part of the input.

under the discrete Fréchet distance is NP-hard to approximate within a factor of $(2 - \varepsilon)$ for $d = 1$ and $(2.25 - \varepsilon)$ for $d \geq 2$ [59]. When the vertices of the sought center curve are restricted to come from the set of all vertices of all input curves, the ℓ-center problem is called middle curve problem [16, 64]. This problem and its variants are NP-hard and W[1]-hard in n [64].

We summarize the main algorithmic results on these problems under the discrete Fréchet distance:

Problem	Approx. Fact.	Running Time	Ambient Space	Reference
ℓ-center	1	$O((mn)^{2\ell} m \log(mn))$	$\mathbb{R}^2$	[61]
	$1 + \varepsilon$	$O((\varepsilon^{-d\ell} + \log m)mn)$	$\mathbb{R}^d$	
	3	$O(nm \log m)$		[59]
ℓ-median	$1 + \varepsilon$	$nm \log^2(m) 2^{O(\varepsilon^{-1} \log \varepsilon^{-1})}$	$\mathbb{R}^d$	[207]
		$O(\varepsilon^{-d\ell} n^2 m)$		[61]
middle curve	1	$O((mn)^{\ell} m \log m)$	$\mathbb{R}^d$	[64]
	$2 + \varepsilon$	$O((\varepsilon^{-d\ell} + \log m)mn)$		

To the best of our knowledge, there are only results on the NP-hardness of the ℓ-median problem in the literature and it is unknown whether the problem is contained in NP or any other complexity class. In the following, we show that the problem is contained in $\exists\mathbb{R}$.

3.1.3 Exact Computation

Here we present a real verification algorithm for deciding Problem 3.1.2, thereby proving that Problem 3.1.2 is contained in $\exists\mathbb{R}$. The idea of the algorithm is to guess the median and the corresponding distances between the input curves and the median and to then verify them using Alt and Godau's algorithm.

Theorem 3.1.3 *The ℓ-median decision problem (Problem 3.1.2) is contained in $\exists\mathbb{R}$.*

Proof. It is easy to see that Problem 3.1.2 can be phrased as a discrete decision problem Q, see Definition 2.5.9. An instance I is structured as follows: the first two integers represent the numerator and the denominator of r, the third integer denotes the number of input curves n and the following n integers $m_1, \ldots, m_n$ represent their complexities. The following $\sum_{i=1}^{n} 2m_i \cdot d$ integers represent the rational coordinates (numerator and denominator) of the input curves. Clearly, $Q(I) = 1$, iff there exists a curve $c \in \mathbb{R}^d_\ell$, such that $\sum_{i=1}^{n} \mathrm{d_F}(\tau_i, c) \leq r$ and else $Q(I) = 0$.

A real verification algorithm A for Q proceeds as follows, where a solution for Q is one integer value, followed by a sequence of up to $d\ell$ real values that represent the complexity and the coordinates of a curve c and n real values $r_1, \ldots, r_n$ that represent the distances between c and the input curves: A checks whether $\sum_{i=1}^{n} r_i \leq r$, if so, it sequentially runs Alt and Godau's algorithm to check whether $\mathrm{d_F}(\tau_i, c) \leq r_i$ for all $i \in [n]$. If any of the tests fails, A rejects, else it accepts.

Clearly, A accepts, if and only if, there is a solution to I. The running time of A is in $O(ndm\ell \log(m\ell))$, thus A fulfills the requirements of Definition 2.5.9. Furthermore, by Theorem 2.5.10 the claim follows. □

On the positive side, Theorem 3.1.3 helps us to narrow down the real complexity of computing a median, which is somewhere in between NP and $\exists\mathbb{R}$. On the negative side, the utilized real verification algorithm is highly nondeterministic, which does not help us to understand the structure of the problem. In the following we aim to reduce the amount of nondeterminism and provide a better understanding of the geometry of the problem. As we will see, the developed approach still depends on nondeterminism and a completely deterministic approach is not in sight.

Assume that we are only provided with the distances between the curves in T and an optimal median c. Based on this knowledge we are able to narrow down the region containing c, see Fig. 3.1. In the following, we show that this information is even sufficient to compute c or another optimal median. For this purpose we define and analyze a generalization of Alt and Godau's free space diagram. We assume that each curve $\tau \in T$ is assigned a radius $r_\tau \in \mathbb{R}_{\geq 0}$ and we use the generalized free space diagram to decide whether there exists a polygonal curve c that simultaneously satisfies $\mathrm{d}_\mathrm{F}(c, \tau) \leq r_\tau$ for all $\tau \in T$ and if existent, compute c. Just as Alt and Godau we solve the problem by deciding whether there exists a polygonal curve in the free space diagram that satisfies certain properties. Since our problem involves n curves, our generalized free space diagram has n dimensions instead of only two.

We note that a related approach has been introduced to compute a mean curve of a set of polygonal curves under the weak Fréchet distance [134]. However, this approach uses product spaces over simplicial complexes in $\mathbb{R}^d$. Another related approach uses an n-dimensional free space diagram to compute the Fréchet distance of a set of curves [90]. This extension of the Fréchet distance can be pictured as the minimum length of a rope that connects n people walking on n curves that may vary in speed without ever going backwards.

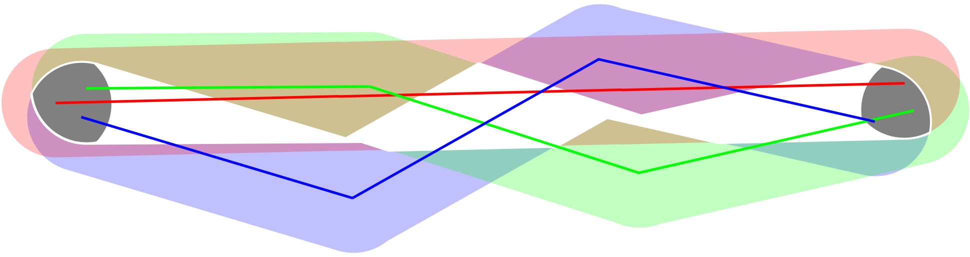

Figure 3.1: Three polygonal curves with enclosed white region that contains the sought median curve. The start-, respectively endpoints, are restricted to lie within the corresponding gray shaded areas inside the white region.

We start by generalizing the cells used to construct the free space diagram. We give a slightly modified definition for two-dimensional cells using the radii r_τ. Note that free space cells are subsets of the two-dimensional Euclidean space.

Definition 3.1.4 *The* ***free space cell*** *of two curves $\sigma, \tau \in T$, $i \in \{2, \dots, |\sigma|\}$ and $j \in \{2, \dots, |\tau|\}$ is the set $\mathcal{C}_{\sigma,\tau,i,j} = \{(\lambda_1, \lambda_2) \in [0,1]^2 \mid \mathcal{I}_{\sigma,\tau,i,j}(\lambda_1, \lambda_2) \neq \emptyset\}$, where*

$$\mathcal{I}_{\sigma,\tau,i,j}(\lambda_1, \lambda_2) = \mathrm{B}\left(\mathrm{lp}\left(\overline{v^\sigma_{i-1}, v^\sigma_i}, \lambda_1\right), r_\sigma\right) \cap \mathrm{B}\left(\mathrm{lp}\left(\overline{v^\tau_{j-1}, v^\tau_j}, \lambda_2\right), r_\tau\right) \subset \mathbb{R}^d$$

is the ***locus*** *of σ, τ, i, j and (λ_1, λ_2), and v^σ_i, respectively v^τ_j, is the i^{th} vertex of σ, respectively j^{th} vertex of τ.*

When we move through such a cell and track the loci of our trajectory, we capture a region that is covered by these loci. This region, if existent, contains the sought curve c by definition.

Definition 3.1.5 *The* ***locus cover*** *of a free space cell $\mathcal{C}_{\sigma,\tau,i,j}$ is the set*

$$\mathcal{L}_{\sigma,\tau,i,j} = \bigcup_{(\lambda_1,\lambda_2)\in\mathcal{C}_{\sigma,\tau,i,j}} \mathcal{I}_{\sigma,\tau,i,j}(\lambda_1,\lambda_2).$$

A locus cover is just a union of loci, which are intersections of two balls of (possibly) different radius. It is easy to see that loci are convex and we prove that the locus covers and the cells themselves are also convex, which makes both easier to handle from both the computational and analytical sight.

Lemma 3.1.6 *For any non-empty set $T \subset \mathbb{R}^d_*$, all $\sigma,\tau \in T$, $i \in \{2,\ldots,|\sigma|\}$ and $j \in \{2,\ldots,|\tau|\}$ the free space cell $\mathcal{C}_{\sigma,\tau,i,j}$, as well as the locus cover $\mathcal{L}_{\sigma,\tau,i,j}$, is a convex set.*

Proof. The condition

$$\mathcal{I}_{\sigma,\tau,i,j}(\lambda_1,\lambda_2) \neq \emptyset$$

is equivalent to $\left\| \mathrm{lp}\left(\overline{v^\sigma_{i-1}, v^\sigma_i}, \lambda_1\right) - \mathrm{lp}\left(\overline{v^\tau_{j-1} v^\tau_j}, \lambda_2\right) \right\| \leq r'$, for $r' = r_\sigma + r_\tau$. Therefore, $\mathcal{C}_{\sigma,\tau,i,j}$ is an r'-free space cell in the sense of Definition 2.7.2 and is convex by Proposition 2.7.3.

For $\lambda \in [0,1]$ we define $b_\sigma(\lambda) = \mathrm{B}\left(\mathrm{lp}\left(\overline{v^\sigma_{i-1}, v^\sigma_i}, \lambda\right), r_\sigma\right)$ and $b_\tau(\lambda) = \mathrm{B}\left(\mathrm{lp}\left(\overline{v^\tau_{j-1}, v^\tau_j}, \lambda\right), r_\tau\right)$ for brevity. By Definitions 3.1.4 and 3.1.5 and by distributivity we have that

$$\begin{aligned}\mathcal{L}_{\sigma,\tau,i,j} &= \bigcup_{(\lambda_1,\lambda_2)\in\mathcal{C}_{\sigma,\tau,i,j}} (b_\sigma(\lambda_1) \cap b_\tau(\lambda_2)) = \bigcup_{\lambda_2\in[0,1]} \bigcup_{\lambda_1\in[0,1]} (b_\sigma(\lambda_1) \cap b_\tau(\lambda_2)) \\ &= \bigcup_{\lambda_2\in[0,1]} \left(\left(\bigcup_{\lambda_1\in[0,1]} b_\sigma(\lambda_1) \right) \cap b_\tau(\lambda_2) \right) = \left(\bigcup_{\lambda_1\in[0,1]} b_\sigma(\lambda_1) \right) \bigcap \left(\bigcup_{\lambda_2\in[0,1]} b_\tau(\lambda_2) \right).\end{aligned}$$

Since $\bigcup_{\lambda_1\in[0,1]} b_\sigma(\lambda_1)$, respectively $\bigcup_{\lambda_2\in[0,1]} b_\tau(\lambda_2)$, is just the convex hull of $b_\sigma(0)$ and $b_\sigma(1)$, respectively $b_\tau(0)$ and $b_\tau(1)$, $\mathcal{L}_{\sigma,\tau,i,j}$ is the intersection of convex sets and is thus convex itself. □

Now, we generalize these definitions to the given set of curves T. Using the intersection of balls instead of pairwise distances is crucial, because three or more balls can have empty intersection, although all pairwise intersections are non-empty. Note that the generalized free space cells are subsets of the n-dimensional Euclidean space.

Definition 3.1.7 *The* ***free space cell*** *of T and $(i_1,\ldots,i_n) \in \prod_{i=1}^{n} \{2,\ldots,|\tau_i|\}$ is the set*

$$\mathcal{C}_{(i_1,\ldots,i_n)} = \{(\lambda_1,\ldots,\lambda_n) \in [0,1]^n \mid \mathcal{I}_{(i_1,\ldots,i_n)}(\lambda_1,\ldots,\lambda_n) \neq \emptyset\},$$

where

$$\mathcal{I}_{(i_1,\ldots,i_n)}(\lambda_1,\ldots,\lambda_n) = \bigcap_{j\in[n]} \mathrm{B}\left(\mathrm{lp}\left(\overline{v^{\tau_j}_{i_j-1} v^{\tau_j}_{i_j}}, \lambda_j\right), r_{\tau_j}\right) \subset \mathbb{R}^d$$

is the ***locus*** *of T, $(i_1,\ldots,i_n)$ and $(\lambda_1,\ldots,\lambda_n)$, and $v^{\tau_j}_{i_j}$ is the $i_j{}^{th}$ vertex of τ_j.*

The generalization of the locus cover is straight-forward. See Fig. 3.2 for a depiction in $\mathbb{R}^2$.

Definition 3.1.8 *The* ***locus cover*** *of a free space cell $\mathcal{C}_{(i_1,\ldots,i_n)}$ is the set*

$$\mathcal{L}_{(i_1,\ldots,i_n)} = \bigcup_{(\lambda_1,\ldots,\lambda_n)\in\mathcal{C}_{(i_1,\ldots,i_n)}} \mathcal{I}_{(i_1,\ldots,i_n)}(\lambda_1,\ldots,\lambda_n).$$

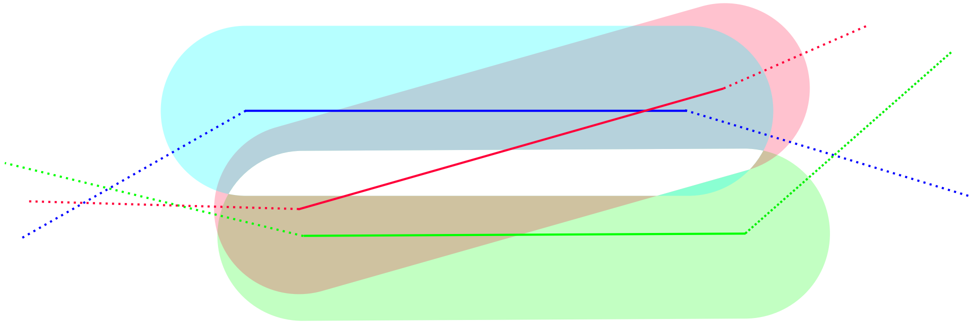

Figure 3.2: The white enclosed region is the locus cover of the free space cell of the three line segments in green, blue and red.

Using Lemma 3.1.6, we prove that the generalized cells and their locus covers are convex sets.

Proposition 3.1.9 *For any non-empty set $T \subset \mathbb{R}^d_*$ and all $(i_1, \dots, i_n) \in \prod_{i=1}^n \{2, \dots, |\tau_i|\}$, the free space cell $\mathcal{C}_{(i_1,\dots,i_n)}$, as well as the locus cover $\mathcal{L}_{(i_1,\dots,i_n)}$, is a convex set.*

Proof. Let $\{\tau_1, \dots, \tau_n\} = T$. We prove the claim by induction on n. For the case $n = 2$, $\mathcal{C}_{\tau_1,\tau_2,i_1,i_2}$ and $\mathcal{L}_{\tau_1,\tau_2,i_1,i_2}$ are convex by Lemma 3.1.6. For $\lambda \in [0,1]$ and $i \in [n]$ we define $b_{\tau_i}(\lambda) = \mathrm{B}\left(\mathrm{lp}\left(\overline{v^{\tau_j}_{i_j-1} v^{\tau_j}_{i_j}}, \lambda\right), r_{\tau_j}\right)$ for brevity, where $v^{\tau_j}_{i_j}$ is the ${i_j}^{\text{th}}$ vertex of τ_j.

Assume the claim holds for $\mathcal{C}_{(i_1,\dots,i_n)}$ and $\mathcal{L}_{(i_1,\dots,i_n)}$. Now, consider $T' = T \cup \{\tau_{n+1}\}$, with $\tau_{n+1} \in \mathbb{R}^d_*$, $i_{n+1} \in \{2, \dots, |\tau_{n+1}|\}$ and $b_{\tau_{n+1}}(\lambda) = \mathrm{B}\left(\mathrm{lp}\left(\overline{v^{\tau_{n+1}}_{i_{n+1}-1} v^{\tau_{n+1}}_{i_{n+1}}}, \lambda\right), r_{\tau_{n+1}}\right)$ for $\lambda \in [0,1]$. Let the corresponding cell and locus cover be denoted $\mathcal{C}_{T',(i_1,\dots,i_{n+1})}$ and $\mathcal{L}_{T',(i_1,\dots,i_{n+1})}$.

By Definitions 3.1.7 and 3.1.8, we obtain

$$\begin{aligned}
\mathcal{L}_{T',(i_1,\dots,i_{n+1})} &= \bigcup_{(\lambda_1,\dots,\lambda_{n+1}) \in \mathcal{C}_{T',(i_1,\dots,i_{n+1})}} \left(\bigcap_{j \in [n+1]} b_{\tau_j}(\lambda_j) \right) \\
&= \bigcup_{(\lambda_1,\dots,\lambda_{n+1}) \in [0,1]^{n+1}} \left(\bigcap_{j \in [n+1]} b_{\tau_j}(\lambda_j) \right) = \bigcap_{j \in [n+1]} \left(\bigcup_{\lambda_j \in [0,1]} b_{\tau_j}(\lambda_j) \right) \\
&= \left(\bigcup_{(\lambda_1,\dots,\lambda_n) \in [0,1]^n} \left(\bigcap_{j \in [n]} b_{\tau_j}(\lambda_j) \right) \right) \bigcap \left(\bigcup_{\lambda_{n+1} \in [0,1]} b_{\tau_{n+1}}(\lambda_{n+1}) \right) \\
&= \left(\bigcup_{(\lambda_1,\dots,\lambda_n) \in C_{(i_1,\dots,i_n)}} \left(\bigcap_{j \in [n]} b_{\tau_j}(\lambda_j) \right) \right) \bigcap \left(\bigcup_{\lambda_{n+1} \in [0,1]} b_{\tau_{n+1}}(\lambda_{n+1}) \right) \\
&= \mathcal{L}_{(i_1,\dots,i_n)} \cap \left(\bigcup_{\lambda_{n+1} \in [0,1]} b_{\tau_{n+1}}(\lambda_{n+1}) \right), \qquad \text{(I)}
\end{aligned}$$

where the third and fourth equation follow from distributivity.

Since $\bigcup_{\lambda_{n+1} \in [0,1]} b_{\tau_{n+1}}(\lambda_{n+1})$ is just the convex hull of $b_{\tau_{n+1}}(0)$ and $b_{\tau_{n+1}}(1)$ and $\mathcal{L}_{(i_1,\dots,i_n)}$ is convex by induction hypothesis, $\mathcal{L}_{T',(i_1,\dots,i_{n+1})}$ is the intersection of convex sets and is thus convex itself.

Now, assume that $\mathcal{C}_{T',(i_1,\dots,i_{n+1})}$ is not convex. Then there exist $(\kappa_1, \dots, \kappa_{n+1}), (\lambda_1, \dots, \lambda_{n+1}) \in \mathcal{C}_{T',(i_1,\dots,i_{n+1})}$ and $t \in [0,1]$, such that $(1-t) \cdot (\kappa_1, \dots, \kappa_{n+1}) + t \cdot (\lambda_1, \dots, \lambda_{n+1}) \notin \mathcal{C}_{T',(i_1,\dots,i_{n+1})}$, but

$(1-t)\cdot(\kappa_1,\dots,\kappa_n)+t\cdot(\lambda_1,\dots,\lambda_n)\in\mathcal{C}_{(i_1,\dots,i_n)}$ by induction hypothesis. Using Eq. (I), this means that $b_{\tau_{n+1}}(\kappa_{n+1})$ and $b_{\tau_{n+1}}(\lambda_{n+1})$ intersect $\mathcal{L}_{(i_1,\dots,i_n)}$, but $b_{\tau_{n+1}}((1-t)\cdot\kappa_{n+1}+t\cdot\lambda_{n+1})$ does not. Let $p\in b_{\tau_{n+1}}(\kappa_{n+1})\cap\mathcal{L}_{(i_1,\dots,i_n)}$ and $q\in b_{\tau_n}(\lambda_{n+1})\cap\mathcal{L}_{(i_1,\dots,i_n)}$. By convexity, $\overline{pq}\subseteq\mathcal{L}_{(i_1,\dots,i_n)}$. Since every $\overline{xy}$ with $x\in b_{\tau_{n+1}}(\kappa_{n+1})$, $y\in b_{\tau_{n+1}}(\lambda_{n+1})$ satisfies $\overline{xy}\cap b_{\tau_{n+1}}((1-t')\cdot\kappa_{n+1}+t'\cdot\lambda_{n+1})\neq\emptyset$ for any $t'\in[0,1]$ by definition, $\overline{pq}\cap b_{\tau_{n+1}}((1-t)\cdot\kappa_{n+1}+t\cdot\lambda_{n+1})\neq\emptyset$ holds, which implies $b_{\tau_{n+1}}((1-t)\cdot\kappa_{n+1}+t\cdot\lambda_{n+1})\cap\mathcal{L}_{(i_1,\dots,i_n)}\neq\emptyset$, a contradiction. □

We can now define the free space diagram, which is a subset of the n-dimensional Euclidean space that uses the free space cells as elementary building blocks.

Definition 3.1.10 *The **free space diagram** of T is the point set*

$$\mathcal{F}=\bigcup_{(i_1,\dots,i_n)\in\prod_{i=1}^{n}\{2,\dots,|\tau_i|\}}\{(\lambda_1+i_1-1,\dots,\lambda_n+i_n-1)\mid(\lambda_1,\dots,\lambda_n)\in\mathcal{C}_{(i_1,\dots,i_n)}\}.$$

We note that in this definition, every two adjacent cells share at least one common point and a point $p=(p_1,\dots,p_n)\in\mathcal{F}$ can be found in the cell with index

$$\iota(p)=(\min\{\lfloor p_1\rfloor+1,|\tau_1|\},\dots,\min\{\lfloor p_n\rfloor+1,|\tau_n|\}).$$

For brevity, we now denote the locus of a point $p=(p_1,\dots,p_n)\in\mathcal{F}$ by

$$\mathcal{I}(p)=\mathcal{I}_{\iota(p)}(\mathbb{1}(p_1<|\tau_1|)\cdot(p_1-\lfloor p_1\rfloor)+\mathbb{1}(p_1\geq|\tau_1|),\dots,\mathbb{1}(p_n<|\tau_n|)\cdot(p_n-\lfloor p_n\rfloor)+\mathbb{1}(p_n\geq|\tau_n|)).$$

Like Alt and Godau, we are interested in computing a certain curve through the free space diagram.

Definition 3.1.11 *For a subset $X\subseteq\mathbb{R}^n$, we define by $\mathbb{C}(X)$ the set of curves $\sigma\colon[0,1]\to X$. By $\mathbb{M}(X)$ we define the subset of curves $\sigma\colon[0,1]\to X$, such that for every $i\in[n]$ the function $\sigma_i\colon[0,1]\to\mathbb{R}$, which denotes the i^{th} component of $\sigma(t)$, is increasing in t.*

We define the reachable free space, which captures the properties of the reparameterizations used in the Fréchet distance.

Definition 3.1.12 *The **reachable free space** of T is the point set*

$$\mathcal{RF}=\{p\in\mathcal{F}\mid\exists\sigma\in\mathbb{M}(\mathcal{F})\exists t\in[0,1]:(\sigma(0)=(1,\dots,1)\wedge\sigma(t)=p)\}.$$

We formally prove our main intuition, which is analogous to the key observation used in Alt and Godau's algorithm [21].

Theorem 3.1.13 *If, and only if, $(|\tau_1|,\dots,|\tau_n|)\in\mathcal{RF}$ then there exists a curve $\sigma\colon[0,1]\to\mathbb{R}^d$, that satisfies $\mathrm{d_F}(\sigma,\tau)\leq r_\tau$ for all $\tau\in T$.*

Proof. If $(|\tau_1|,\dots,|\tau_n|)\in\mathcal{RF}$, let $\gamma\in\mathbb{M}(\mathcal{F})$ be a curve in $\mathcal{F}$ with $\gamma(0)=(1,\dots,1)$ and $\gamma(1)=(|\tau_1|,\dots,|\tau_n|)$. γ corresponds to a closed region in $\mathbb{R}^d$, i.e., the union of the loci of the points of γ. This region must contain the sought curve σ by Definitions 3.1.7 and 3.1.8.

Now, for $p\in\mathcal{F}$ we define

$$f(p)=\text{lexicographically smallest point in }\mathcal{I}(p).$$

Since the lexicographic order on the Cartesian product over reals is a total order (see e.g. [138]), $\mathcal{I}(p)$ is a compact set and $p \in \mathcal{F}$, such a unique element exists and thus $f(p)$ is well-defined.

For $i \in [n]$, let $v_1^{\tau_i}, \dots v_{|\tau_i|}^{\tau_i}$ be the vertices, of τ_i. Let $V = \max_{i \in [n]} \max_{j \in \{2,\dots,|\tau_i|\}} \|v_j^{\tau_i} - v_{j-1}^{\tau_i}\|$. Now, by Definition 3.1.7 we have for any two points $p, q \in \mathcal{F}$ that $\frac{\|f(p)-f(q)\|}{\|p-q\|}$ is maximal iff moving from p to q translates the border of a ball determining $f(p)$ to $f(q)$, such that the center of this ball traverses a maximum length edge. Therefore, $\frac{\|f(p)-f(q)\|}{\|p-q\|} \leq V$ and hence f is V-Lipschitz (see Definition 2.2.4).

Since γ and f are continuous in their respective domains, the function $\sigma = f \circ \gamma$ is continuous, too. σ is the curve we are looking for. We now prove that σ has distance at most r_τ to every $\tau \in T$. For every $i \in [n]$, by definition of γ we have $\mathrm{d_F}(\sigma, \tau_i) \leq \max_{t \in [0,1]} \|\sigma(t) - \tau_i'(t)\| \leq r_{\tau_i}$, where

$$\tau_i'(t) = \begin{cases} \tau_i(0), & \text{if } t = 0 \\ \mathrm{lp}\left(\overline{v^{\tau_i}_{\lceil \gamma_i(t) \rceil - 1} v^{\tau_i}_{\lceil \gamma_i(t) \rceil}}, \lambda_i(t)\right), & \text{else,} \end{cases}$$

$\gamma_i(t)$ denotes the i^{th} component of $\gamma(t)$ and $\lambda_i(t) = \gamma_i(t) - \lceil \gamma_i(t) \rceil + 1$. Namely, τ_i' and τ_i represent the same curve but under different parameterizations, where the parameterization of τ_i' is determined by γ and adheres to the restrictions imposed by the Fréchet distance by definition of γ.

If $(|\tau_1|, \dots, |\tau_n|) \notin \mathcal{RF}$, either there is no $\gamma \in \mathbb{C}(\mathcal{F})$ with $\gamma(0) = (1, \dots, 1)$ and $\gamma(1) = (|\tau_1|, \dots, |\tau_n|)$. Therefore by Definition 3.1.7 there is at least one $\tau \in T$ and a $t \in [0,1]$ such that there is no point $p \in \mathbb{R}^d$ with

- $\|\tau(t) - p\| \leq r_\tau$ and
- for all $\tau' \in T \setminus \{\tau\}$ there is a $t' \in [0,1]$ such that $\|\tau'(t) - p\| \leq r_{\tau'}$,

hence there is no curve σ that satisfies $\mathrm{d_F}(\tau, \sigma) \leq r_\tau$ for all $\tau \in T$. Or all curves $\gamma \in \mathbb{C}(\mathcal{F})$ with $\gamma(0) = (1, \dots, 1)$ and $\gamma(1) = (|\tau_1|, \dots, |\tau_n|)$ are not contained in $\mathbb{M}(\mathcal{F})$, therefore σ can not exist by Definitions 2.3.8 and 2.3.11. □

By definition the curve σ must not be polygonal and it is generally not a viable idea to compute it, since one might not find a compact representation for it[2]. However, Theorem 3.1.13 enlightens the key concepts that can be used to derive the desired result. Therefore, in the following we aim at computing a polygonal curve of bounded complexity with similar properties. We capture this in the following theorem.

Theorem 3.1.14 *Assume that there exists a polygonal curve $\gamma \in \mathbb{M}(\mathcal{F})$ with $|\gamma| = \ell$, $\gamma(0) = (1, \dots, 1)$ and $\gamma(1) = (|\tau_1|, \dots, |\tau_n|)$. Let $v_1^\gamma, \dots, v_\ell^\gamma$ be the vertices of γ.*

For all polygonal curves $\sigma \in \mathbb{R}^d_\ell$ with vertices $v_1^\sigma \in \mathcal{I}(v_1^\gamma), \dots, v_\ell^\sigma \in \mathcal{I}(v_\ell^\gamma)$ it holds that $\mathrm{d_F}(\sigma, \tau) \leq r_\tau$ for all $\tau \in T$.

Proof. Let f be the function that maps v_i^γ to v_i^σ for all $i \in [\ell]$. We follow the argumentation of the proof of Theorem 3.1.13 and show that we can extend f to a continuous function from $\{\gamma(t) \mid t \in [0,1]\}$ to $\{\sigma(t) \mid t \in [0,1]\}$, such that $f \circ \gamma = \sigma$ and thus $\mathrm{d_F}(\sigma, \tau) \leq r_\tau$ for all $\tau \in T$.

Assume now that we divide γ into line segments $\overline{p_1 p_2}, \dots, \overline{p_{k-1} p_k}$, so that for each $j \in [k-1]$ there exists a tuple $(i_1, \dots, i_n) \in \prod_{i=1}^n \{2, \dots, |\tau_i|\}$, such that $\overline{p_j p_{j+1}} \subseteq \mathcal{C}_{(i_1, \dots, i_n)}$. The required

[2] One may represent it as a spline, but the number of polynomial pieces might be large.

extension of f might not be possible, if the convex hull of $\mathcal{I}(p_j)$ and $\mathcal{I}(p_{j+1})$, denoted by H_j, is not contained in the union of loci determined by $\overline{p_j p_{j+1}}$, denoted $I_j = \bigcup_{t\in[0,1]} \mathcal{I}(\operatorname{lp}(\overline{p_j p_{j+1}}, t))$, for any $j \in [k-1]$.

Assume for a $j \in [k-1]$, this is the case, i.e., $H_j \not\subseteq I_j$. Then there is a $t \in (0,1)$, $q_1 \in \mathcal{I}(p_j)$ and $q_2 \in \mathcal{I}(p_{j+1})$ such that $s = \operatorname{lp}(\overline{q_1 q_2}, t) = (1-t)q_1 + tq_2 \notin I_j$. Let $\mathcal{C}_{(i_1,\dots,i_n)}$ be the cell that contains $\overline{p_j p_{j+1}}$ and for $a \in [n]$ let $\lambda_{j,a} = p_{j,a} - \lfloor p_{j,a} \rfloor$, $\lambda_{j+1,a} = p_{j+1,a} - \lfloor p_{j+1,a} \rfloor$, $u_a = \operatorname{lp}\left(\overline{v_{i_a-1}^{\tau_a} v_{i_a}^{\tau_a}}, \lambda_{j,a}\right)$ and $w_a = \operatorname{lp}\left(\overline{v_{i_a-1}^{\tau_a} v_{i_a}^{\tau_a}}, \lambda_{j+1,a}\right)$, where $p_{j,a}$, respectively $p_{j+1,a}$, is the a^{th} component of p_j, respectively p_{j+1}, and $v_{i_a}^{\tau_a}$, respectively $v_{i_a+1}^{\tau_a}$, is the ${i_a}^{\text{th}}$, respectively $(i_a+1)^{\text{th}}$, vertex of τ_a. From Definition 3.1.7 we know that $\|q_1 - u_a\| \leq r_{\tau_a}$ and $\|q_2 - w_a\| \leq r_{\tau_a}$ for each $a \in [n]$. Further, we obtain for each $a \in [n]$:

$$\begin{aligned}
x_a &= \operatorname{lp}\left(\overline{v_{i_a-1}^{\tau_a} v_{i_a}^{\tau_a}}, (1-t)\lambda_{j,a} + t\lambda_{j+1,a}\right) \\
&= (1 - [(1-t)\lambda_{j,a} + t\lambda_{j+1,a}])v_{i_a-1}^{\tau_a} + [(1-t)\lambda_{j,a} + t\lambda_{j+1,a}]v_{i_a}^{\tau_a} \\
&= (1 - \lambda_{j,a} + t\lambda_{j,a} - t\lambda_{j+1,a})v_{i_a-1}^{\tau_a} + (\lambda_{j,a} - t\lambda_{j,a} + t\lambda_{j+1,a})v_{i_a}^{\tau_a} \\
&= (1 - t - \lambda_{j,a} + t\lambda_{j,a})v_{i_a-1}^{\tau_a} + (\lambda_{j,a} - t\lambda_{j,a} + t\lambda_{j+1,a})v_{i_a}^{\tau_a} + t(1-\lambda_{j+1,a})v_{i_a-1}^{\tau_a} \\
&= (1-t)[(1-\lambda_{j,a})v_{i_a-1}^{\tau_a} + \lambda_{j,a}v_{i_a}^{\tau_a}] + t[(1-\lambda_{j+1,a})v_{i_a-1}^{\tau_a} + \lambda_{j+1,a}v_{i_a}^{\tau_a}] \\
&= (1-t)u_a + tw_a.
\end{aligned}$$

Therefore, by the triangle inequality $\|s - x_a\| = \|[(1-t)q_1 + tq_2] - [(1-t)u_a + tw_a]\| \leq (1-t)\|q_1 - u_a\| + t\|q_2 - w_a\| \leq r_{\tau_a}$ and hence $s \in I_j$, a contradiction. As a consequence, we can extend f to a continuous function that maps every point in $\{\gamma(t) \mid t \in [0,1]\}$ to a certain point of $\{\sigma(t) \mid t \in [0,1]\}$. Let $t_1^\gamma, \dots, t_\ell^\gamma$ be the instants of γ and $t_1^\sigma, \dots, t_\ell^\sigma$ be the instants of σ. For $i \in [\ell-1]$ and $p \in \{\gamma(t) \mid t \in (t_i^\gamma, t_{i+1}^\gamma)\}$ we define $f(p) = \sigma(g_i(p))$, where $g_i(p) = \min\{t \in (t_i^\sigma, t_{i+1}^\sigma) \mid \sigma(t) \in \mathcal{I}(p)\}$. Since $\mathcal{I}(p)$ is a compact set and by definition γ has no self-intersections, each g_i is well-defined and therefore f is also well-defined.

Let $V = \max_{j\in\{2,\dots,|\sigma|\}} \|v_j^\sigma - v_{j-1}^\sigma\|$. Analogously to the argumentation in the proof of Theorem 3.1.13 it can be shown that f is V-Lipschitz, hence $f \circ \gamma = \sigma$ is continuous and consequently $\mathrm{d_F}(\sigma,\tau) \leq r_\tau$ for all $\tau \in T$, which yields the claim. □

Now that we have gained some insights on the geometry of the problem through Proposition 3.1.9 and Theorems 3.1.13 and 3.1.14, we formulate the rough procedure that an algorithm, which is less dependent on nondeterminism than the algorithm described in Theorem 3.1.3, and that utilizes these insights may follow.

Polygonal Chain Stabbing

Since free space cells are convex, analogously to Alt and Godau's approach, we focus on their borders. Namely, we want to compute a monotonic polygonal curve γ of complexity at most ℓ that starts in $(0,\dots,0)$, ends in $(|\tau_1|,\dots,|\tau_n|)$ and visits an admissible sequence of borders of the *reachable* free space cells while always staying within the convex hull of two consecutive borders, where a sequence of borders is *admissible*, if for every two consecutive borders there exists a cell that contains them. If we have computed γ, by Theorem 3.1.14 any curve that has its vertices in the loci of γ's vertices is a median curve c (for suitable radii $r_{\tau_1},\dots,r_{\tau_n}$). If we can compute a suitable representation of these loci, we can compute the median c.

For the sake of completeness, we now formally define these borders as follows.

Definition 3.1.15 *The j^{th}* ***outgoing border****, where $j \in [n]$, of a cell $\mathcal{C}_{(i_1,\ldots,i_n)}$ is the set*

$$\partial_j \mathcal{C}_{(i_1,\ldots,i_n)} = \{(\lambda_1 + i_1 - 1, \ldots, \lambda_n + i_n - 1) \mid (\lambda_1, \ldots, \lambda_n) \in \mathcal{C}_{(i_1,\ldots,i_n)}, \lambda_j = 1\}.$$

See Fig. 3.3 for a depiction of the loci of such a border.

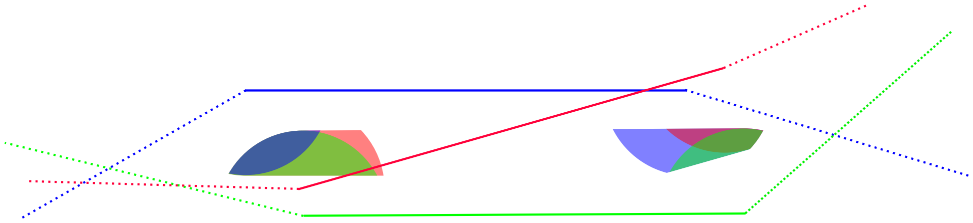

Figure 3.3: Loci corresponding to incoming and outgoing borders of a cell, colored with respect to the corresponding line segment.

A similar setting has already been studied for convex objects in $\mathbb{R}^2$ [94, 125]. Here, a polygonal curve is called a polygonal chain and a polygonal curve or line segment that visits (intersects) an object is said to stab it. Since polygonal curves and line segments (when we interpret these as polygonal curves of complexity two) are directed, an order of visitation is induced.

The central idea in these approaches is to compute a so-called wedge data structure for a maximum length sub-sequence of objects that can be visited by a line segment in the order of the sequence. This idea originates from Melkman and O'Rourke [199] and has later been extended by Guibas et al. [125]. A wedge data structure yields a description of *all* line segments that stab a given sequence of convex objects in order and Guibas et al. extend this idea to a chain-stabbing wedge, whereas the former they call line-stabbing wedge and use these as a building block.

Now, an algorithm for computing such a curve γ could compute for every admissible sequence of outgoing borders a polygonal chain stabbing wedge, thereby checking if the complexity restriction is met, and then check if the wedge contains a monotonic curve that starts in $(0, \ldots, 0)$ and ends in $(|\tau_1|, \ldots, |\tau_n|)$. Of course this approach comes with a high running time, since there are up to $(2^{n-2})^{\sum_{i=1}^{n}(|\tau_i|-1)} < 2^{n^2 m}$ admissible sequences of borders.

Unfortunately, the approach in [125] only works for certain convex objects in $\mathbb{R}^2$ and our cell borders are "complex" $(n-1)$-dimensional convex objects in $\mathbb{R}^n$. For computing loci there is also an existing approach [221]. However, this approach is also limited. It only works for balls of equal radius in $\mathbb{R}^3$ and while polygonal curves in $\mathbb{R}^3$ are a common input, the restriction to equal radii is not acceptable, though.

In summary, we have found a potentially viable approach to compute a median curve c, when provided with the optimal radii $r_{\tau_1} = \mathrm{d_F}(c, \tau_1), \ldots, r_{\tau_n} = \mathrm{d_F}(c, \tau_n)$. The running time of this approach can already be bounded as $\Omega(2^{2n^2-2n})$. However, there are still open problems: extending the approaches of Guibas et al. and Ramos.

3.1.4 Randomized Approximation Algorithms

We have gained geometrical insights into the problem, but failed to design a deterministic algorithm that solves the problem exactly. This is not very surprising, since the problem is related to the geometric median problem (for curves of complexity one), for which no exact algorithm exists (under standard models of computation). Since the problem is also NP-hard, we now focus on efficient approximation algorithms. In the following, we formulate randomized approximation algorithms, which we successively improve by developing some central ideas.

Prelude

We describe a simple sampling scheme that can be applied to any data set from a metric space and yields a good approximation in reasonable time. However, a drawback is that the approximate median stems from the data set itself and therefore does not adhere to the complexity restriction on the median curve. In detail, we draw two uniform samples from the input. The first sample contains so-called *candidates* and the second sample contains so-called *witnesses*. The candidates contain a good approximate median with high probability and the witnesses are used to evaluate the cost of each candidate. Finally, the candidate that evaluates best against the witnesses is returned. We formally prove that this certain candidate is a good approximate median with high probability.

Proposition 3.1.16 *Given a finite set $T \subset \mathbb{R}^d_m$, for any $\varepsilon, \delta \in (0, 1/2)$ we can use a uniform sample S of cardinality $O(\ln(1/\delta)/\varepsilon)$ of candidates and a uniform sample W of cardinality $O(\ln(|S|/\delta)/\varepsilon^2)$ of witnesses, to obtain with probability at least $1-\delta$ a $(2+\varepsilon)$-approximate ℓ-median for T with up to m vertices.*

Proof. Let $c^* \in \arg\min_{c \in \mathbb{R}^d_\ell} \sum_{\tau \in T} \mathrm{d_F}(c, \tau)$ be an optimal ℓ-median for T. Since S is a uniform sample and by linearity we have $\mathrm{Exp}[\mathrm{d_F}(s, c^*)] = \frac{1}{|T|} \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*)$, for any $s \in S$. Now, let

$$B_{1+\varepsilon} = \left\{ \tau \in T \mid \mathrm{d_F}(\tau, c^*) \leq \frac{(1+\varepsilon)}{|T|} \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*) \right\}.$$

For any $\sigma \in B_{1+\varepsilon}$ by the triangle inequality it holds that

$$\sum_{\tau \in T} \mathrm{d_F}(\tau, \sigma) \leq \sum_{\tau \in T} (\mathrm{d_F}(\tau, c^*) + \mathrm{d_F}(c^*, \sigma)) \leq (2+\varepsilon) \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*).$$

Thus, it holds that $\mathrm{cost}(T, \sigma) \leq (2+\varepsilon)\,\mathrm{cost}(T, c^*)$. For $i \in [|S|]$, let F_i denote the event that $s_i \notin B_{1+\varepsilon}$. By Markov's inequality we have that $\Pr[F_i] \leq \frac{1}{1+\varepsilon} < 1$.

Further, by independence and by choosing $|S| \geq \left\lceil \frac{2\ln(2/\delta)}{\varepsilon} \right\rceil$ the probability that $B_{1+\varepsilon} \cap S = \emptyset$ is

$$\Pr\left[\bigcap_{i=1}^{|S|} F_i\right] \leq \frac{1}{(1+\varepsilon)^{|S|}} \leq \frac{1}{\exp(\frac{\varepsilon}{2}|S|)} \leq \exp(-\varepsilon \ln(2/\delta)/\varepsilon) = \frac{\delta}{2}.$$

Let $c^*_S \in \arg\min_{s \in S} \sum_{\tau \in T} \mathrm{d_F}(\tau, s)$. We do not want any bad candidate $t \in S$ with $\sum_{\tau \in T} \mathrm{d_F}(\tau, t) > (1+\varepsilon) \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*_S)$ to have $\sum_{w \in W} \mathrm{d_F}(w, t) \leq \sum_{w \in W} \mathrm{d_F}(w, c^*_S)$. By Theorem 2.4.19, a union bound over the elements of S and by choosing $|W| \geq \frac{64}{\varepsilon^2} \ln(2|S|/\delta)$, the probability for this event is bounded by

$$|S| \exp\left(-\frac{\varepsilon^2 |W|}{64}\right) \leq |S| \exp\left(-\ln\left(2|S|/\delta\right)\right) \leq \frac{\delta}{2}.$$

Now, if we take the $s \in S$ that minimizes $\sum_{w \in W} \mathrm{d_F}(w, s)$, by a union bound, with probability at least $1-\delta$ it holds that

$$\sum_{\tau \in T} \mathrm{d_F}(\tau, s) \leq (1+\varepsilon) \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*_S) \leq (1+\varepsilon)(2+\varepsilon) \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*) \leq (2+4\varepsilon) \sum_{\tau \in T} \mathrm{d_F}(\tau, c^*).$$

The claim follows by rescaling ε by $\frac{1}{4}$. □

A great benefit is that this algorithm can be implemented to run in time $O\left(\ln\left(\frac{\ln(1/\delta)}{\delta\varepsilon}\right)\varepsilon^{-2}m^2\log m\right)$ using two for-loops (one over the candidates and one over the witnesses) and Alt and Godau's algorithm to compute the distances. For this reason, in the following we refine and extend this idea.

Simple and Fast 34-Approximation

Here, we present a 34-approximation algorithm for the ℓ-median problem which builds upon the ideas of Proposition 3.1.16. To fix the problem with the complexity of the resulting curve, we simplify the candidate $s \in S$ that evaluated best against the witnesses W using an efficient minimum-error ℓ-simplification approximation algorithm. Though this downgrades the approximation factor, Algorithm 3 is very fast in terms of the input size. Indeed, it has worst-case running time logarithmic in n and sub-quartic in m.

Algorithm 3 ℓ-Median by Simplification

```
procedure ℓ-MEDIAN-34-APPROXIMATION($T = \{\tau_1, \dots, \tau_n\}, \delta$)
    $S \leftarrow$ sample $\lceil 2(\ln(2) - \ln(\delta)) \rceil$ curves from $T$ uniformly and independently
        with replacement
    $\gamma \leftarrow \lceil -64(\ln(\delta) - \ln(\lceil 4(\ln(2) - \ln(\delta)) \rceil)) \rceil$
    $W \leftarrow$ sample $\gamma$ curves from $T$ uniformly and independently with replacement
    $t \leftarrow$ arbitrary elem. from $\arg\min_{s \in S} \mathrm{cost}(W, s)$
    return $\mathrm{simpl}(\ell, \alpha, t)$                                  ▷ [21, 149]
```

Next, we prove the quality of approximation of Algorithm 3.

Theorem 3.1.17 *Given a parameter $\delta \in (0,1)$ and a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, Algorithm 3 returns with probability at least $1 - \delta$ a polygonal curve $c \in \mathbb{R}^d_\ell$, such that $\mathrm{cost}(T, c^*) \leq \mathrm{cost}(T, c) \leq (6 + 7\alpha) \cdot \mathrm{cost}(T, c^*)$, where c^* is an optimal ℓ-median for T and α is the approximation factor of the utilized minimum-error ℓ-simplification approximation algorithm.*

Proof. First, we know that $\mathrm{d_F}(\tau, \mathrm{simpl}(\ell, \alpha, \tau)) \leq \alpha \cdot \mathrm{d_F}(\tau, c^*)$, for each $\tau \in T$, by Definition 2.8.5.

Now, there are at least $\frac{n}{2}$ curves in T that are within distance at most $\frac{2\,\mathrm{cost}(T,c^*)}{n}$ to c^*. Otherwise, the cost of the remaining curves would exceed $\mathrm{cost}(T, c^*)$, which is a contradiction. Hence, each $s \in S$ has probability at least $\frac{1}{2}$ to be within distance $\frac{2\,\mathrm{cost}(T,c^*)}{n}$ to c^*.

Since the elements of S are sampled independently we conclude that the probability that every $s \in S$ has distance to c^* greater than $\frac{2\,\mathrm{cost}(T,c^*)}{n}$ is at most $(1 - \frac{1}{2})^{|S|} \leq \exp\left(-\frac{2(\ln(2) - \ln(\delta))}{2}\right) = \frac{\delta}{2}$.

Now, assume there is a $s \in S$ with $\mathrm{d_F}(s, c^*) \leq \frac{2\,\mathrm{cost}(T,c^*)}{n}$. We do not want any $t \in S \setminus \{s\}$ with $\mathrm{cost}(T, t) > 2\,\mathrm{cost}(T, s)$ to have $\mathrm{cost}(W, t) \leq \mathrm{cost}(W, s)$. Using Theorem 2.4.19 we conclude that this happens with probability at most

$$\exp\left(-\frac{-64(\ln(\delta) - \ln(\lceil 4(\ln(2) - \ln(\delta)) \rceil))}{64}\right) \leq \frac{\delta}{\lceil 4(\ln(2) - \ln(\delta)) \rceil} \leq \frac{\delta}{2|S|},$$

for each $t \in S \setminus \{s\}$.

Using a union bound over all bad events, we conclude that with probability at least $1-\delta$, Algorithm 3 samples a curve $s \in S$, with $\mathrm{d_F}(s,c^*) \leq 2\operatorname{cost}(T,c^*)/n$ and returns the simplification $c = \operatorname{simpl}(\ell,\alpha,t)$ of a curve $t \in S$, with $\operatorname{cost}(T,t) \leq 2\operatorname{cost}(T,s)$. The triangle inequality yields

$$\sum_{\tau\in T}(\mathrm{d_F}(t,c^*) - \mathrm{d_F}(\tau,c^*)) \leq \sum_{\tau\in T}\mathrm{d_F}(t,\tau) \leq 2\sum_{\tau\in T}\mathrm{d_F}(s,\tau) \leq 2\sum_{\tau\in T}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,s)),$$

which is equivalent to

$$n\cdot\mathrm{d_F}(t,c^*) \leq 2\operatorname{cost}(T,c^*) + \operatorname{cost}(T,c^*) + 2n\frac{2\operatorname{cost}(T,c^*)}{n} \iff \mathrm{d_F}(t,c^*) \leq \frac{7\operatorname{cost}(T,c^*)}{n}.$$

Hence, we have

$$\begin{aligned}\operatorname{cost}(T,c) &= \sum_{\tau\in T}\mathrm{d_F}(\tau,\operatorname{simpl}(\ell,\alpha,t)) \leq \sum_{\tau\in T}(\mathrm{d_F}(\tau,t) + \mathrm{d_F}(t,\operatorname{simpl}(\ell,\alpha,t)))\\ &\leq 2\operatorname{cost}(T,s) + \sum_{\tau\in T}\alpha\cdot\mathrm{d_F}(t,c^*) \leq 2\sum_{\tau\in T}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,s)) + 7\alpha\cdot\operatorname{cost}(T,c^*)\\ &\leq 2\operatorname{cost}(T,c^*) + 4\operatorname{cost}(T,c^*) + 7\alpha\cdot\operatorname{cost}(T,c^*) = (6+7\alpha)\operatorname{cost}(T,c^*).\end{aligned}$$

The lower bound $\operatorname{cost}(T,c^*) \leq \operatorname{cost}(T,c)$ follows from the fact that the returned curve has ℓ vertices and that c^* has minimum cost among all curves with ℓ vertices. □

Combining Theorem 3.1.17 and Theorem 2.8.6, we obtain the following corollary.

Corollary 3.1.18 *Given a parameter $\delta \in (0,1)$ and a set $T \subset \mathbb{R}^d_m$ of polygonal curves, Algorithm 3 returns with probability at least $1-\delta$ a polygonal curve $c \in \mathbb{R}^d_\ell$, such that*

$$\operatorname{cost}(T,c^*) \leq \operatorname{cost}(T,c) \leq 34\cdot\operatorname{cost}(T,c^*),$$

where c^* is an optimal ℓ-median for T, in time $O(m^2\log(m)\ln^2(1/\delta) + m^3\log m)$, when the algorithms by Imai and Iri [149] and Alt and Godau [21] are combined for ℓ-simplification.

Proof. We use Theorem 2.8.6 together with Theorem 3.1.17, which yields an approximation factor of 34.

Now, drawing the samples takes time $O(\ln(1/\delta))$ each. Evaluating the samples against each other takes time $O(m^2\log(m)\ln^2(1/\delta))$ and simplifying one of the curves that evaluates best takes time $O(m^3\log m)$. We conclude that Algorithm 3 has running time $O(m^2\log(m)\ln^2(1/\delta) + m^3\log m)$. □

An Outlook to the (k,ℓ)-Median Problem

In Chapter 4 we present an approximation algorithm for (k,ℓ)-median clustering. In fact, this algorithm is more general, it can approximate every *generalized k-median clustering problem*, when provided with a problem-specific plugin algorithm for solving/approximating the corresponding median problem. Explanation and formal details are provided in Section 4.2. Here, it suffices to know that any algorithm serving as such a plugin is required to have an additional input $\beta \in [1,\infty)$ and needs to give a (probabilistic) guarantee that its return value, which may comprise a set of items, contains an (approximate) median for an arbitrary fixed subset T' of the input that has size at least a β-fraction of the input size.

The following algorithms are designed to fulfill these requirements, such that they can be used as the required plugin. However, if we set $\beta = 1$ these algorithms can also be used to approximate the ℓ-median problem. All we have to do is to evaluate the candidates that the algorithms return against the input and return the best one.

$(3+\varepsilon)$-Approximation by Simple Shortcutting

Here, we present an algorithm that returns candidates, containing with high probability a $(3+\varepsilon)$-approximate ℓ-median of complexity at most $2\ell - 2$ for a subset that takes a constant fraction of the input.

To achieve a better approximation factor than that of Algorithm 3, we want to avoid simplification. We note that the approximation factor of Algorithm 3 can be improved – for example by using a larger candidate sample, but not by much. Instead of simplification we want to discretize the regions containing the median's vertices by well-adjusted grids and compute a good approximate median by enumeration and validation, which is a basic approach in geometric approximation algorithms. Since we are working with the continuous variant of the Fréchet distance there is a considerable drawback: the vertices of an ℓ-median c^* do not have to be located in the balls of radius $r = \mathrm{d_F}(\tau, c^*)$, centered at an input curve τ's vertices – due to the continuous nature they can be located anywhere with distance r to an edge.

We circumvent this by introducing so-called shortcutting lemmata. Shortcutting is a technique, which has already been used in the literature [54, 82, 83], and whose underlying idea is simple: replace a sub-curve by the line segment determined by its start point and end point. It has mainly been used for partial curve matching under the Fréchet distance, but it also shows to be very effective for our purposes, when shortcuts are carefully introduced. We start with a simple lemma, which states that we can indeed search the aforementioned balls if we accept a resulting curve of complexity at most $2\ell - 2$. See Fig. 3.4 for a visualization.

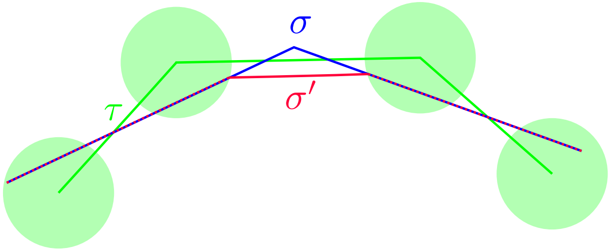

Figure 3.4: τ is a curve from the input that is close to an optimal median σ. By inserting a shortcut we can find a curve σ' that has the same distance to τ as σ but with all vertices contained in the balls of radius $\mathrm{d_F}(\tau, \sigma)$ centered at τ's vertices.

Lemma 3.1.19 *Let $\sigma, \tau \in \mathbb{R}^d_*$ be polygonal curves. Let $v^\tau_1, \dots, v^\tau_{|\tau|}$ be the vertices of τ and let $r = \mathrm{d_F}(\sigma, \tau)$. There exists a polygonal curve $\sigma' \in \mathbb{R}^d_{2|\sigma|-2}$ with $\mathrm{d_F}(\sigma', \tau) \leq \mathrm{d_F}(\sigma, \tau)$ and every vertex contained in at least one of $B(v^\tau_1, r), \dots, B(v^\tau_{|\tau|}, r)$.*

Proof. Let $v^\sigma_1, \dots, v^\sigma_{|\sigma|}$ be the vertices of σ. Further, let $t^\sigma_1, \dots, t^\sigma_{|\sigma|}$ and $t^\tau_1, \dots, t^\tau_{|\tau|}$ be the instants of σ and τ, respectively. Also, for $h \in \mathcal{H}$ (recall that $\mathcal{H}$ is the set of all continuous bijections $h\colon [0,1] \to [0,1]$ with $h(0) = 0$ and $h(1) = 1$), let $r_h = \max\limits_{t \in [0,1]} \lVert \sigma(t) - \tau(h(t)) \rVert$ be the distance realized by h. We know from Proposition 2.3.12 that there exists a sequence $(h_x)_{x=1}^{\infty}$ in $\mathcal{H}$, such that $\lim\limits_{x\to\infty} r_{h_x} = \mathrm{d_F}(\sigma, \tau) = r$.

Now, fix an arbitrary $h \in \mathcal{H}$ and assume there is a vertex v_i^σ of σ, with instant t_i^σ, that is not contained in any of $B(v_1^\tau, r_h), \dots, B(v_{|\tau|}^\tau, r_h)$. Let j be the maximum of $[|\tau| - 1]$, such that $t_j^\tau \leq h(t_i^\sigma) \leq t_{j+1}^\tau$. So v^σ is matched to $\overline{\tau(t_j^\tau)\tau(t_{j+1}^\tau)}$ by h. We modify σ in such a way, that v_i^σ is replaced by two new vertices that are elements of $B(v_j^\tau, r_h)$ and $B(v_{j+1}^\tau, r_h)$, respectively.

Namely, let t^- be the maximum of $[0, t_i^\sigma)$, such that $\sigma(t^-) \in B(v_j^\tau, r_h)$ and let t^+ be the minimum of $(t_i^\sigma, 1]$, such that $\sigma(t^+) \in B(v_{j+1}^\tau, r_h)$. These are the instants when σ leaves $B(v_j^\tau, r_h)$ before visiting v_i^σ and σ enters $B(v_{j+1}^\tau, r_h)$ after visiting v_i^σ, respectively. Let σ_h' be the piecewise defined curve, defined just like σ on $[0, t^-]$ and $[t^+, 1]$, but on (t^-, t^+) it connects $\sigma(t^-)$ and $\sigma(t^+)$ with the line segment $s(t) = \left(1 - \frac{t-t^-}{t^+-t^-}\right)\tau(t^-) + \frac{t-t^-}{t^+-t^-}\tau(t^+)$.

We know that $\|\sigma(t^-) - \tau(h(t^-))\| \leq r_h$ and $\|\sigma(t^+) - \tau(h(t^+))\| \leq r_h$. Note that $t_j^\tau \leq h(t^-)$ and $h(t^+) \leq t_{j+1}^\tau$ since $\sigma(t^-)$ and $\sigma(t^+)$ are the closest points to v_i^σ on σ that have distance r_h to v_j^τ and v_{j+1}^τ, respectively, by definition. Therefore, τ has no vertices between the instants $h(t^-)$ and $h(t^+)$. Now, h can be used to match $\sigma_h'|_{[0,t^-)}$ to $\tau|_{[0,h(t^-))}$ and $\sigma_h'|_{(t^+,1]}$ to $\tau|_{(t^+,1]}$ with distance at most r_h. Since $\sigma_h'|_{[t^-,t^+]}$ and $\tau|_{[h(t^-),h(t^+)]}$ are just line segments, they can be linearly matched to each other with distance at most $\max\{\|\sigma_h'(t^-) - \tau(h(t^-))\|, \|\sigma_h'(t^+) - \tau(h(t^+))\|\} \leq r_h$. We conclude that $\mathrm{d_F}(\sigma_h', \tau) \leq r_h$.

Because this modification works for every $h \in \mathcal{H}$, we have $\mathrm{d_F}(\sigma_h', \tau) \leq r_h$ for every $h \in \mathcal{H}$. Thus, $\lim\limits_{x\to\infty} \mathrm{d_F}(\sigma_{h_x}', \tau) \leq \mathrm{d_F}(\sigma, \tau) = r$.

Now, to prove the claim, for every $h \in \mathcal{H}$ we apply this modification to v_i^σ and successively to every other vertex $v_i^{\sigma_h'}$ of the resulting curve σ_h', not contained in one of the balls, until every vertex of σ_h' is contained in a ball. Note that the modification is repeated at most $|\sigma| - 2$ times for every $h \in \mathcal{H}$, since the start and end vertex of σ must be contained in $B(v_1^\tau, r_h)$ and $B(v_{|\tau|}^\tau, r_h)$, respectively. Therefore, the number of vertices of every σ_h' can be bounded by $2 \cdot (|\sigma| - 2) + 2$ since every other vertex must not lie in a ball and for each such vertex one new vertex is created. Thus, $|\sigma_h'| \leq 2|\sigma| - 2$. □

We now present Algorithm 4, which works similar as Algorithm 3, but uses shortcutting instead of simplification. As a consequence, we can achieve an approximation factor of $3 + \varepsilon$ instead of a factor of $(2 + \varepsilon)(1 + \alpha)$ (where $(2 + \varepsilon)$ comes from the candidate sampling and $(1 + \alpha)$ comes from simplification with approximation factor $\alpha \geq 1$). Indeed, this factor is the best we can achieve by the previously used techniques in combination with simplification. Thus, to achieve an approximation factor of $(4 + \varepsilon)$ one would need to compute the optimal minimum-error ℓ-simplifications of the input curves and to the best of our knowledge, there is no such algorithm for the continuous Fréchet distance.

Algorithm 4 utilizes the superset sampling technique (see Section 2.4.1) to fulfill the requirements to be used as plugin algorithm for the (k, ℓ)-median approximation algorithm to be presented in Chapter 4. Therefore, it has running time exponential in the size of the sample S. A further difference is that we need an upper and a lower bound on the cost of an optimal ℓ-median c^* for T', to properly set up the grids we use for shortcutting. The lower bound can be obtained by simple estimation, using Markov's inequality – with high probability a multiple of the cost of the result of Algorithm 3 run on a subset $S' \subseteq S$ with $S' \subseteq T'$ and with respect to S' is a lower bound on the cost of c^* with respect to T'. For the upper bound we utilize a case distinction, which guarantees us that if we fail to obtain an upper bound on the optimal cost, the result of Algorithm 3 then is a good approximation (factor $2+\varepsilon$, an immediate consequence of the distinction) and can be used instead of a best curve obtained by shortcutting.

Algorithm 4 has several parameters: β determines the size (in terms of a fraction of the input) of the smallest subset of the input for which an approximate median can be computed, δ determines the probability of failure of the algorithm and ε determines the approximation factor.
The algorithm first draws a sample S from the whole input and then loops over all subsets $S' \subseteq S$ of certain size to find an S' that is a sample from the subset $T' \subseteq T$. For any possible S' the lower and upper bound are computed, the grids are set up and all possible curves are enumerated. Since the algorithm can not judge which set S' is a sample from T', all possible curves are returned as candidates.

Algorithm 4 ℓ-Median for Subset by Simple Shortcutting

```
procedure ℓ-MEDIAN-(3 + ε)-CANDIDATES($T = \{\tau_1, \dots, \tau_n\}, \beta, \delta, \varepsilon$)
    $\varepsilon' \leftarrow \varepsilon/3$, $C \leftarrow \emptyset$
    $S \leftarrow$ sample $\lceil -8\beta(\varepsilon')^{-1}(\ln(\delta) - \ln(4)) \rceil$ curves from $T$ uniformly and independently
        with replacement
    for $S' \subseteq S$ with $|S'| = \frac{|S|}{2\beta}$ do
        $c \leftarrow$ ℓ-MEDIAN-34-APPROXIMATION($S', \delta/4$) (Algorithm 3)
        $\Delta \leftarrow \mathrm{cost}(S', c)$, $\Delta_l \leftarrow \frac{\delta n}{2|S|} \frac{\Delta}{34}$, $\Delta_u \leftarrow \frac{1}{\varepsilon'}\Delta$, $C \leftarrow C \cup \{c\}$
        for $s \in S'$ do
            $P \leftarrow \emptyset$
            for $i \in [|s|]$ do
                $P \leftarrow P \cup \mathbb{G}\left(B\left(v_i^s, (1+\varepsilon')\Delta_u\right), \frac{\varepsilon'}{n\sqrt{d}}\Delta_l\right)$        ▷ $v_i^s$: $i^{\text{th}}$ vertex of $s$
            $C \leftarrow C \cup$ set of all polygonal curves with $2\ell - 2$ vertices from $P$
    return $C$
```

We prove the quality of approximation of Algorithm 4.

Theorem 3.1.20 *Given three parameters $\beta \in [1, \infty)$, $\delta, \varepsilon \in (0,1)$ and a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, with probability at least $1-\delta$ the set of candidates that Algorithm 4 returns contains a $(3+\varepsilon)$-approximate ℓ-median with up to $2\ell - 2$ vertices for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$.*

Proof. We assume that $|T'| \geq \frac{1}{\beta}|T|$. Let n' be the number of sampled curves in S that are elements of T'. Clearly, $\mathrm{Exp}[n'] \geq \sum_{i=1}^{|S|} \frac{1}{\beta} = \frac{|S|}{\beta}$. Also, n' is the sum of independent Bernoulli trials. A Chernoff bound (see Theorem 2.4.17) yields:

$$\Pr\left[n' < \frac{|S|}{2\beta}\right] \leq \Pr\left[n' < \frac{1}{2}\mathrm{Exp}[n']\right] \leq \exp\left(-\frac{1}{4}\frac{|S|}{2\beta}\right) \leq \exp\left(\frac{\ln(\delta) - \ln(4)}{\varepsilon}\right) = \left(\frac{\delta}{4}\right)^{\frac{1}{\varepsilon}} \leq \frac{\delta}{4}.$$

In other words, with probability at most $\delta/4$ no subset $S' \subseteq S$, of cardinality at least $\frac{|S|}{2\beta}$, is a subset of T'. We condition the rest of the proof on the contrary event, denoted by $\mathcal{E}_{T'}$, namely, that there is a subset $S' \subseteq S$ with $S' \subseteq T'$ and $|S'| \geq \frac{|S|}{2\beta}$. Note that S' is then a uniform and independent sample of T' (see Section 2.4.1).

Now, let $c^* \in \arg\min_{c \in \mathbb{R}^d_\ell} \operatorname{cost}(T', c)$ be an optimal ℓ-median for T'. The expected distance between $s \in S'$ and c^* is

$$\operatorname{Exp}[\mathrm{d_F}(s, c^*) \mid \mathcal{E}_{T'}] = \sum_{\tau \in T'} \mathrm{d_F}(c^*, \tau) \cdot \frac{1}{|T'|} = \frac{\operatorname{cost}(T', c^*)}{|T'|}.$$

By linearity, we have $\operatorname{Exp}[\operatorname{cost}(S', c^*) \mid \mathcal{E}_{T'}] = \frac{|S'|}{|T'|} \operatorname{cost}(T', c^*)$. Markov's inequality yields:

$$\Pr\left[\frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) > \operatorname{cost}(T', c^*) \;\middle|\; \mathcal{E}_{T'}\right] \leq \frac{\delta}{4}.$$

We conclude that with probability at most $\delta/4$ we have $\frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) > \operatorname{cost}(T', c^*)$.

Using Markov's inequality again, for every $s \in S'$ we have

$$\Pr\left[\mathrm{d_F}(s, c^*) > (1+\varepsilon)\frac{\operatorname{cost}(T', c^*)}{|T'|} \;\middle|\; \mathcal{E}_{T'}\right] \leq \frac{1}{1+\varepsilon},$$

therefore by independence

$$\Pr\left[\min_{s \in S'} \mathrm{d_F}(s, c^*) > (1+\varepsilon)\frac{\operatorname{cost}(T', c^*)}{|T'|} \;\middle|\; \mathcal{E}_{T'}\right] \leq \frac{1}{(1+\varepsilon)^{|S'|}} \leq \exp\left(-\frac{\varepsilon}{2}\frac{|S|}{2\beta}\right).$$

Hence, with probability at most $\exp\left(-\frac{\varepsilon\left\lceil -\frac{8\beta(\ln(\delta)-\ln(4))}{\varepsilon}\right\rceil}{4\beta}\right) \leq \delta^2/16 \leq \delta/4$ there is no $s \in S'$ with $\mathrm{d_F}(s, c^*) \leq (1+\varepsilon)\frac{\operatorname{cost}(T', c^*)}{|T'|}$. Also, with probability at most $\delta/4$ Algorithm 3 fails to compute a 34-approximate ℓ-median $c \in \mathbb{R}^d_\ell$ for S', cf. Corollary 3.1.18.

Using Proposition 2.4.5, we conclude that with probability at least $1-\delta$ all the following events occur simultaneously:

1. There is a subset $S' \subseteq S$ of cardinality at least $|S|/(2\beta)$ that is a uniform and independent sample of T',
2. there is a curve $s \in S'$ with $\mathrm{d_F}(s, c^*) \leq (1+\varepsilon)\frac{\operatorname{cost}(T', c^*)}{|T'|}$,
3. Algorithm 3 computes a polygonal curve $c \in \mathbb{R}^d_\ell$ with $\operatorname{cost}(S', c^*_{S'}) \leq \operatorname{cost}(S', c) \leq 34 \operatorname{cost}(S', c^*_{S'})$, where $c^*_{S'} \in \mathbb{R}^d_\ell$ is an optimal ℓ-median for S',
4. and it holds that $\frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) \leq \operatorname{cost}(T', c^*)$.

Since $c^*_{S'}$ is an optimal ℓ-median for S' we get the following from the last two items:

$$\operatorname{cost}(T', c^*) \geq \frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) \geq \frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*_{S'}) \geq \frac{\delta|T'|}{4|S'|} \frac{\operatorname{cost}(S', c)}{34}.$$

We now distinguish between two cases:

Case 1: $\mathrm{d_F}(c, c^*) \geq (1+2\varepsilon)\frac{\operatorname{cost}(T', c^*)}{|T'|}$

The triangle inequality yields

$$\begin{aligned}\mathrm{d_F}(c,s) \geq \mathrm{d_F}(c,c^*) - \mathrm{d_F}(c^*,s) &\geq \mathrm{d_F}(c,c^*) - (1+\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|}\\ &\geq (1+2\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|} - (1+\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|} = \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}.\end{aligned}$$

As a consequence, $\operatorname{cost}(S',c) \geq \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} \iff \frac{\operatorname{cost}(T',c^*)}{|T'|} \leq \frac{1}{\varepsilon}\operatorname{cost}(S',c)$.

Now, let $v_1^s,\dots,v_{|s|}^s$ be the vertices of s. By Lemma 3.1.19 there exists a polygonal curve c' with up to $2\ell-2$ vertices, every vertex contained in one of $B(v_1^s,\mathrm{d_F}(c^*,s)),\dots,B(v_{|s|}^s,\mathrm{d_F}(c^*,s))$ and $\mathrm{d_F}(s,c') \leq \mathrm{d_F}(s,c^*) \leq (1+\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|} \leq (1+\varepsilon)\frac{\operatorname{cost}(S',c)}{\varepsilon}$.

In the set of candidates, that Algorithm 4 returns, a curve c'' with up to $2\ell-2$ vertices from the union of the grid covers and distance at most $\frac{\varepsilon\frac{\delta n}{2|S|}\operatorname{cost}(S',c)}{n} \leq \frac{\varepsilon\frac{\delta|T'|}{4|S'|}\operatorname{cost}(S',c)}{|T'|} \leq \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}$ between every corresponding pair of vertices of c' and c'' is contained. We conclude that $\mathrm{d_F}(c',c'') \leq \frac{\varepsilon\operatorname{cost}(T',c^*)}{|T'|}$.

We can now bound the cost of c'' as follows:

$$\begin{aligned}\operatorname{cost}(T',c'') = \sum_{\tau\in T'}\mathrm{d_F}(\tau,c'') &\leq \sum_{\tau\in T'}\left(\mathrm{d_F}(\tau,c') + \frac{\varepsilon\operatorname{cost}(T',c^*)}{|T'|}\right)\\ &\leq \sum_{\tau\in T'}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,c')) + \varepsilon\operatorname{cost}(T,c^*)\\ &\leq \sum_{\tau\in T'}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,s) + \mathrm{d_F}(s,c')) + \varepsilon\operatorname{cost}(T',c^*) \leq (3+3\varepsilon)\operatorname{cost}(T',c^*).\end{aligned}$$

Case 2: $\mathrm{d_F}(c,c^*) < (1+2\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|}$

The cost of c can easily be bounded:

$$\operatorname{cost}(T',c) \leq \sum_{\tau\in T'}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,c)) < \operatorname{cost}(T',c^*) + (1+2\varepsilon)\operatorname{cost}(T',c^*) = (2+2\varepsilon)\operatorname{cost}(T',c^*).$$

The claim follows by rescaling ε by $\frac{1}{3}$. □

Next we analyze the worst-case running time of Algorithm 4 and the number of candidates it returns.

Theorem 3.1.21 *The running time as well as the number of candidates that Algorithm 4 returns is in* $2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}+\log(m)\right)}$.

Proof. The sample S has size $O\left(\frac{\ln(1/\delta)\cdot\beta}{\varepsilon}\right)$ and sampling it takes time $O\left(\frac{\ln(1/\delta)\cdot\beta}{\varepsilon}\right)$. Let $n_S = |S|$. The outer for-loop runs

$$\binom{n_S}{\frac{n_S}{2\beta}} \in 2^{O\left(\frac{n_S}{2\beta}\log n_S\right)} \subset 2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}\right)}$$

times. In each iteration, we run Algorithm 3, taking time $O(m^2 \log(m) \ln^2(1/\delta) + m^3 \log m)$ (cf. Corollary 3.1.18), we compute the cost of the returned curve with respect to S', taking time $O\left(\frac{\ln(1/\delta)}{\varepsilon'} \cdot m \log(m)\right)$, and per curve in S' we build up to m grids of size

$$\left(\frac{\frac{(1+\varepsilon')\Delta}{\varepsilon'}}{\frac{\varepsilon'\delta n\Delta}{n\sqrt{d}2|S|34}}\right)^d = \left(\frac{68\sqrt{d}|S|(1+\varepsilon')}{\varepsilon'^2\delta}\right)^d \in O\left(\frac{\beta^d(\ln(1/\delta))^d}{\varepsilon^{3d}\delta^d}\right)$$

each. For each curve $s \in S'$, Algorithm 4 then enumerates all combinations of $2\ell - 2$ points from these up to m grids, resulting in

$$O\left(\frac{m^{2\ell-2}\beta^{2\ell d-2d}(\ln(1/\delta))^{2\ell d-2d}}{\varepsilon^{6\ell d-6d}\delta^{2\ell d-2d}}\right)$$

candidates per $s \in S'$, per iteration of the for-loop.

Thus, Algorithm 4 computes $O\left(\text{poly}(m, \beta, \delta^{-1}, \varepsilon^{-1})\right)$ candidates per iteration of the for-loop and enumeration also takes time $O\left(\text{poly}(m, \beta, \delta^{-1}, \varepsilon^{-1})\right)$ per iteration of the for-loop (where $\text{poly}(x_1, x_2, \ldots)$ denotes a polynomial function in $x_1, x_2, \ldots$).

All in all, we have running time and number of candidates $2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}+\log(m)\right)}$. □

Since each candidate returned by the algorithm can be evaluated against the input in time $O(nm \log m)$ using Alt and Godau's algorithm, the following corollary follows.

Corollary 3.1.22 *There exists an algorithm that, given a parameter $\varepsilon \in (0,1)$ and a set $T = \{\tau_1, \ldots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, returns with constant positive probability a $(3+\varepsilon)$-approximate ℓ-median with up to $2\ell - 2$ vertices for T in time $n \cdot 2^{O(\varepsilon^{-2}+\log m)}$.*

$(1+\varepsilon)$-Approximation by Advanced Shortcutting

Next, we present an algorithm that returns candidates, containing with high probability a $(1+\varepsilon)$-approximate ℓ-median of complexity at most $2\ell - 2$ for a subset that takes a constant fraction of the input. Before we present the algorithm, we present our second shortcutting lemma. Now, we do not introduce shortcuts with respect to a single curve, but with respect to several curves: by introducing shortcuts with respect to $\varepsilon|T|$ well-chosen curves from the input T, for a given $\varepsilon \in (0,1)$, we preserve the distances to at least $(1-\varepsilon)|T|$ curves from T. In this context well-chosen means that there exists a certain number of subsets of T, of each we have to pick a curve for shortcutting. This will enable the high quality of approximation of Algorithm 5, which we formalize in the following lemma.

Lemma 3.1.23 *Let $\sigma \in \mathbb{R}^d_*$ be a polygonal curve with $|\sigma| > 2$ vertices and $T = \{\tau_1, \ldots, \tau_n\} \subset \mathbb{R}^d_*$ be a set of polygonal curves. For $i \in [n]$, let $r_i = \mathrm{d_F}(\tau_i, \sigma)$ and for $j \in [|\tau_i|]$, let $v^{\tau_i}_j$ be the j^{th} vertex of τ_i. For any $\varepsilon \in (0,1)$ there are $2|\sigma| - 4$ subsets $T_1, \ldots, T_{2|\sigma|-4} \subseteq T$, not necessarily disjoint, and of $\frac{\varepsilon n}{2|\sigma|}$ curves each, such that for every subset $T' \subseteq T$ containing at least one curve out of each $T_k \in \{T_1, \ldots, T_{2|\sigma|-4}\}$, a polygonal curve $\sigma' \in \mathbb{R}^d_{2|\sigma|-2}$ exists with every vertex contained in*

$$\bigcup_{\tau_i \in T'} \bigcup_{j \in [|\tau_i|]} B(v^{\tau_i}_j, r_i)$$

and $\mathrm{d_F}(\tau, \sigma') \leq \mathrm{d_F}(\tau, \sigma)$ for each $\tau \in T \setminus (T_1 \cup \cdots \cup T_{2|\sigma|-4})$.

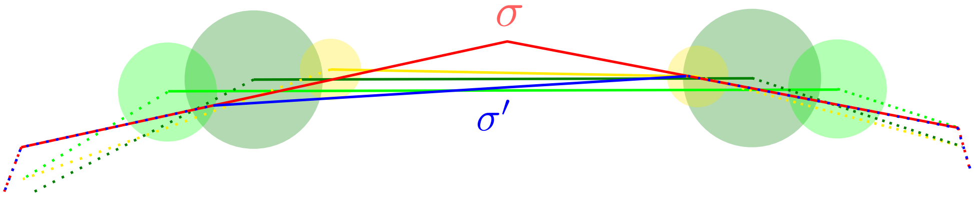

Figure 3.5: By using a subset of well-chosen input curves, a shortcut can be constructed that preserves the majority of distances to the input curves: $\mathrm{d_F}(\sigma', \tau) \leq \mathrm{d_F}(\sigma, \tau)$ for most $\tau \in T$.

The idea is the following, see Fig. 3.5 for a visualization. One can argue that every vertex v of σ not contained in any of the balls centered at the vertices of the curves in T (and of radius according to their distance to σ) can be shortcut by connecting the last point p^- before v (in terms of the parameter of σ) contained in one ball and first point p^+ after v contained in one ball. This does not increase the Fréchet distances between σ and the $\tau \in T$, because only matchings among line segments are affected by this modification. Furthermore, most distances are preserved when we do not actually use the last and first ball before and after v, but one of the $\frac{\varepsilon n}{2|\sigma|}$ balls before and one of the $\frac{\varepsilon n}{2|\sigma|}$ balls after v, which is the key of the following proof.

Proof of Lemma 3.1.23. Let $\ell = |\sigma|$. For the sake of simplicity, we assume that $\frac{\varepsilon n}{2\ell}$ is integral. For $i \in [n]$, let $v_1^{\tau_i}, \dots, v_{|\tau_i|}^{\tau_i}$ be the vertices of τ_i with instants $t_1^{\tau_i}, \dots, t_{|\tau_i|}^{\tau_i}$ and let $v_1^{\sigma}, \dots, v_{\ell}^{\sigma}$ be the vertices of σ with instants $t_1^{\sigma}, \dots, t_{\ell}^{\sigma}$. Also, for $h \in \mathcal{H}$ (recall that $\mathcal{H}$ is the set of all continuous bijections $h \colon [0,1] \to [0,1]$ with $h(0) = 0$ and $h(1) = 1$) and $i \in [n]$, let $r_{i,h} = \max_{t \in [0,1]} \|\sigma(t) - \tau_i(h(t))\|$ be the distance realized by h with respect to τ_i. We know from Proposition 2.3.12 that for each $i \in [n]$ there exists a sequence $(h_{i,x})_{x=1}^{\infty}$ in $\mathcal{H}$, such that $\lim_{x \to \infty} r_{i,h_{i,x}} = \mathrm{d_F}(\sigma, \tau_i) = r_i$.

In the following, given arbitrary $h_1, \dots, h_n \in \mathcal{H}$, we describe how to modify σ, such that its vertices can be found in the balls around the vertices of the $\tau \in T$, of radii determined by $h_1, \dots, h_n$. Later we will argue that this modification can be applied using the $h_{1,x}, \dots, h_{n,x}$, for each $x \in \mathbb{N}$, in particular.

Now, fix arbitrary $h_1, \dots, h_n \in \mathcal{H}$ and for an arbitrary $k \in \{2, \dots, |\sigma| - 1\}$, fix the vertex v_k^{σ} of σ with instant t_k^{σ}. For $i \in [n]$, let s_i be the maximum of $[|\tau_i| - 1]$, such that $t_{s_i}^{\tau_i} \leq h_i(t_k^{\sigma}) \leq t_{s_i+1}^{\tau_i}$. Namely, v_k^{σ} is matched to a point on the line segment $\overline{v_{s_1}^{\tau_1} v_{s_1+1}^{\tau_1}}, \dots, \overline{v_{s_n}^{\tau_n} v_{s_n+1}^{\tau_n}}$, respectively, by $h_1, \dots, h_n$.

For $i \in [n]$, let t_i^- be the maximum of $[0, t_k^{\sigma}]$, such that $\sigma(t_i^-) \in B(v_{s_i}^{\tau_i}, r_{i,h_i})$ and let t_i^+ be the minimum of $[t_k^{\sigma}, 1]$, such that $\sigma(t_i^+) \in B(v_{s_i+1}^{\tau_i}, r_{i,h_i})$. These are the instants when σ visits $B(v_{s_i}^{\tau_i}, r_{i,h_i})$ before or when it visits v_k^{σ} and σ visits $B(v_{s_i+1}^{\tau_i}, r_{i,h_i})$ when or after it visits v_k^{σ}, respectively. Furthermore, there is a permutation $\alpha \in \mathcal{S}_n$ of the index set $[n]$, such that

$$t_{\alpha^{-1}(1)}^- \leq \dots \leq t_{\alpha^{-1}(n)}^-. \tag{I}$$

Also, there is a permutation $\zeta \in \mathcal{S}_n$ of the index set $[n]$, such that

$$t_{\zeta^{-1}(1)}^+ \leq \dots \leq t_{\zeta^{-1}(n)}^+. \tag{II}$$

Additionally, for each $i \in [n]$ we have

$$t_{s_i}^{\tau_i} \leq h_i(t_i^-) \tag{III}$$

and

$$h_i(t_i^+) \leq t^{\tau_i}_{s_i+1}, \tag{IV}$$

because $\sigma(t_i^-)$ and $\sigma(t_i^+)$ are the closest points to v^σ on σ that have distance at most r_{i,h_i} to $v^{\tau_i}_{s_i}$ and $v^{\tau_i}_{s_i+1}$, respectively, by definition. We will now use Eqs. (I) to (IV) to prove that an advanced shortcut only affects matchings among line segments and hence we can easily bound the resulting distances for at least $(1-\varepsilon)n$ of the curves.

Let

$$I_{v^\sigma_k}(h_1,\dots,h_n) = \{\tau_{\alpha^{-1}((1-\frac{\varepsilon}{2\ell})n+1)}, \dots, \tau_{\alpha^{-1}(n)}\},\ O_{v^\sigma_k}(h_1,\dots,h_n) = \{\tau_{\zeta^{-1}(1)}, \dots, \tau_{\zeta^{-1}(\frac{\varepsilon n}{2\ell})}\}.$$

$I_{v^\sigma_k}(h_1,\dots,h_n)$ is the set of the last $\frac{\varepsilon n}{2\ell}$ curves whose balls are visited by σ, before or when σ visits v^σ_k. Similarly, $O_{v^\sigma_k}(h_1,\dots,h_n)$ is the set of the first $\frac{\varepsilon n}{2\ell}$ curves whose balls are visited by σ, when or immediately after σ visited v^σ_k. We now modify σ, such that v^σ_k is replaced by two new vertices that are elements of at least one $B(v^{\tau_i}_j, r_{i,h_i})$, for a $\tau_i \in I_{v^\sigma_k}(h_1,\dots,h_n)$, respectively for a $\tau_i \in O_{v^\sigma_k}(h_1,\dots,h_n)$, and $j \in [|\tau_i|]$, each.

Let $\sigma'_{h_1,\dots,h_n}$ be the piecewise defined curve, defined just like σ on $\left[0, t^-_{\alpha^{-1}(k_1)}\right]$ and $\left[t^+_{\zeta^{-1}(k_2)}, 1\right]$ for arbitrary $k_1 \in \{(1-\frac{\varepsilon}{2\ell})n+1,\dots,n\}$ and $k_2 \in [\frac{\varepsilon n}{2\ell}]$, but on $\left(t^-_{\alpha^{-1}(k_1)}, t^+_{\zeta^{-1}(k_2)}\right)$ it connects $\sigma\left(t^-_{\alpha^{-1}(k_1)}\right)$ and $\sigma\left(t^+_{\zeta^{-1}(k_2)}\right)$ with the line segment

$$\gamma(t) = \left(1 - \frac{t - t^-_{\alpha^{-1}(k_1)}}{t^+_{\zeta^{-1}(k_2)} - t^-_{\alpha^{-1}(k_1)}}\right)\sigma\left(t^-_{\alpha^{-1}(k_1)}\right) + \frac{t - t^-_{\alpha^{-1}(k_1)}}{t^+_{\zeta^{-1}(k_2)} - t^-_{\alpha^{-1}(k_1)}}\sigma\left(t^+_{\zeta^{-1}(k_2)}\right).$$

We now argue that for all $\tau_i \in T \setminus (I_{v^\sigma_k}(h_1,\dots,h_n) \cup O_{v^\sigma_k}(h_1,\dots,h_n))$ the Fréchet distance between $\sigma'_{h_1,\dots,h_n}$ and τ_i is upper bounded by r_{i,h_i}. First, note that by definition $h_1,\dots,h_n$ are strictly increasing functions, since they are continuous bijections that map 0 to 0 and 1 to 1. As immediate consequence, we have that

$$t^{\tau_i}_{s_i} \leq h_i(t_i^-) \leq h_i\left(t^-_{\alpha^{-1}(k_1)}\right) \tag{V}$$

for each $\tau_i \in T \setminus I_{v^\sigma_k}(h_1,\dots,h_n)$ and

$$h_i\left(t^+_{\zeta^{-1}(k_2)}\right) \leq h_i(t_i^+) \leq t^{\tau_i}_{s_i+1} \tag{VI}$$

for each $\tau_i \in T \setminus O_{v^\sigma_k}(h_1,\dots,h_n)$, using Eqs. (I) to (IV). Therefore, each $\tau_i \in T \setminus (I_{v^\sigma_k}(h_1,\dots,h_n) \cup O_{v^\sigma_k}(h_1,\dots,h_n))$ has no vertex between the instants $h_i\left(t^-_{\alpha^{-1}(k_1)}\right)$ and $h_i\left(t^+_{\zeta^{-1}(k_2)}\right)$. We also know that for each $\tau_i \in T$

$$\left\|\sigma\left(t^-_{\alpha^{-1}(k_1)}\right) - \tau_i\left(h_i\left(t^-_{\alpha^{-1}(k_1)}\right)\right)\right\| \leq r_{i,h_i} \tag{VII}$$

and

$$\left\|\sigma\left(t^+_{\zeta^{-1}(k_2)}\right) - \tau_i\left(h_i\left(t^+_{\zeta^{-1}(k_2)}\right)\right)\right\| \leq r_{i,h_i}. \tag{VIII}$$

Let $D_{s,\sigma} = \left[0, t^-_{\alpha^{-1}(k_1)}\right)$, $D_{m,\sigma} = \left[t^-_{\alpha^{-1}(k_1)}, t^+_{\zeta^{-1}(k_2)}\right]$ and $D_{e,\sigma} = \left(t^+_{\zeta^{-1}(k_2)}, 1\right]$. Also, for $i \in [n]$, let $D_{s,\tau_i} = \left[0, h_i\left(t^-_{\alpha^{-1}(k_1)}\right)\right)$, $D_{m,\tau_i} = \left[h_i\left(t^-_{\alpha^{-1}(k_1)}\right), h_i\left(t^+_{\zeta^{-1}(k_2)}\right)\right]$ and $D_{e,\tau_i} = \left(h_i\left(t^+_{\zeta^{-1}(k_2)}\right), 1\right]$. Now, for each $\tau_i \in T \setminus (I_{v^\sigma_k}(h_1,\dots,h_n) \cup O_{v^\sigma_k}(h_1,\dots,h_n))$ we use h_i to match $\sigma'_{h_1,\dots,h_n}|_{D_{s,\sigma}}$ to $\tau_i|_{D_{s,\tau_i}}$ and $\sigma'_{h_1,\dots,h_n}|_{D_{e,\sigma}}$ to $\tau_i|_{D_{e,\tau_i}}$ with distance at most r_{i,h_i}. Since $\sigma'_{h_1,\dots,h_n}|_{D_{m,\sigma}}$ and $\tau_i|_{D_{m,\tau_i}}$

are just line segments by Eqs. (V) and (VI), they can be linearly matched to each other with distance at most

$$\max\left\{\left\|\sigma\left(t^-_{\alpha^{-1}(k_1)}\right) - \tau_i\left(h_i\left(t^-_{\alpha^{-1}(k_1)}\right)\right)\right\|, \left\|\sigma\left(t^+_{\zeta^{-1}(k_2)}\right) - \tau_i\left(h_i\left(t^+_{\zeta^{-1}(k_2)}\right)\right)\right\|\right\},$$

which is at most r_{i,h_i} by Eqs. (VII) and (VIII). We conclude that $\mathrm{d_F}(\sigma'_{h_1,\dots,h_n}, \tau_i) \leq r_{i,h_i}$.

Because this modification works for every $h_1, \dots, h_n \in \mathcal{H}$, we conclude that $\mathrm{d_F}(\sigma'_{h_1,\dots,h_n}, \tau_i) \leq r_{i,h_i}$ for every $h_1, \dots, h_n \in \mathcal{H}$ and $\tau_i \in T \setminus (I_{v_k^\sigma}(h_1, \dots, h_n) \cup O_{v_k^\sigma}(h_1, \dots, h_n))$. Thus,

$$\lim_{x\to\infty} \mathrm{d_F}(\sigma'_{h_{1,x},\dots,h_{n,x}}, \tau_i) \leq \mathrm{d_F}(\sigma, \tau_i) = r_i \text{ for each } \tau_i \in T \setminus (I_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x}) \cup O_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x})).$$

Now, to prove the claim, for each combination $h_1, \dots, h_n \in \mathcal{H}$, we apply this modification to v_k^σ and successively to every other vertex $v_l^{\sigma'_{h_1,\dots,h_n}}$ of the resulting curve $\sigma'_{h_1,\dots,h_n}$, except $v_1^{\sigma'_{h_1,\dots,h_n}}$ and $v_{|\sigma'_{h_1,\dots,h_n}|}^{\sigma'_{h_1,\dots,h_n}}$, since these must be elements of $B(v_1^{\tau_i}, r_{i,h_i})$ and $B(v_{|\tau_i|}^{\tau_i}, r_{i,h_i})$, respectively, for each $i \in [n]$, by definition of the Fréchet distance.

Since the modification is repeated at most $|\sigma| - 2$ times for each combination $h_1, \dots h_n \in \mathcal{H}$, we conclude that the number of vertices of each $\sigma'_{h_1,\dots,h_n}$ can be bounded by $2 \cdot (|\sigma| - 2) + 2$.

$T_1, \dots, T_{2\ell-4}$ are therefore all the $I_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x})$ and $O_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x})$ for $k \in \{2, \dots, 2|\sigma| - 3\}$, when $x \to \infty$. Note that every $I_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x})$ and $O_{v_k^\sigma}(h_{1,x}, \dots, h_{n,x})$ is determined by the visiting order of the balls and since their radii converge, these sets do too. □

We now present Algorithm 5, which is nearly identical to Algorithm 4 but uses the advanced shortcutting lemma. In detail, we have to draw a larger sample S to use the advanced shortcutting and consider the union of all grid points over all curves from S' for enumerating curves.
Like Algorithm 4, this algorithm can be used as plugin in the recursive k-median approximation-scheme (Algorithm 9) that we present in Section 4.2.

Algorithm 5 ℓ-Median for Subset by Advanced Shortcutting

```
procedure ℓ-MEDIAN-(1+ε)-CANDIDATES($T = \{\tau_1, \dots, \tau_n\}, \beta, \delta, \varepsilon$)
    $\varepsilon' \leftarrow \varepsilon/6$, $C \leftarrow \emptyset$
    $S \leftarrow$ sample $\lceil -8\beta\ell(\varepsilon')^{-1}(\ln(\delta) - \ln(4(2\ell-4)))\rceil$ curves from $T$
            uniformly and independently with replacement
    for $S' \subseteq S$ with $|S'| = \frac{|S|}{2\beta}$ do
        $c \leftarrow$ ℓ-MEDIAN-34-APPROXIMATION($S', \delta/4$) (Algorithm 3)
        $\Delta \leftarrow \mathrm{cost}(S', c)$, $\Delta_l \leftarrow \frac{\delta n}{2|S|}\frac{\Delta}{34}$, $\Delta_u \leftarrow \frac{1}{\varepsilon'}\Delta$
        $C \leftarrow C \cup \{c\}$, $P \leftarrow \emptyset$
        for $s \in S'$ do
            for $i \in [|s|]$ do
                $P \leftarrow P \cup \mathbb{G}\left(B\left(v_i^s, \frac{4\ell}{\varepsilon'}\Delta_u\right), \frac{\varepsilon'}{n\sqrt{d}}\Delta_l\right)$        ▷ $v_i^s$: $i^{\text{th}}$ vertex of $s$
        $C \leftarrow C \cup$ set of all polygonal curves with $2\ell - 2$ vertices from $P$
    return $C$
```

We prove the quality of approximation of Algorithm 5.

Theorem 3.1.24 *Given three parameters $\beta \in [1, \infty)$, $\delta \in (0, 1)$, $\varepsilon \in (0, 0.158]$ and a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, with probability at least $1 - \delta$ the set of candidates that Algorithm 5 returns contains a $(1 + \varepsilon)$-approximate ℓ-median with up to $2\ell - 2$ vertices for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$.*

In the following proof we make use of a case distinction developed by Nath and Taylor [207, Proof of Theorem 10], which is a key ingredient to enable the $(1 + \varepsilon)$-approximation, though the domain of ε has to be restricted to $(0, 0.158]$.

Proof of Theorem 3.1.24. We assume that $|T'| \geq \frac{1}{\beta}|T|$ and $\ell > 2$. Let n' be the number of sampled curves in S that are elements of T'. Clearly, $\operatorname{Exp}[n'] \geq \sum_{i=1}^{|S|} \frac{1}{\beta} = \frac{|S|}{\beta}$. Also, n' is the sum of independent Bernoulli trials. A Chernoff bound (cf. Theorem 2.4.17) yields:

$$\Pr\left[n' < \frac{|S|}{2\beta}\right] \leq \Pr\left[n' < \frac{\operatorname{Exp}[n']}{2}\right] \leq \exp\left(-\frac{1}{4}\frac{|S|}{2\beta}\right) \leq \exp\left(\frac{\ell \ln\left(\frac{\delta}{4(2\ell-4)}\right)}{\varepsilon}\right) \leq \left(\frac{\delta^\ell}{8^\ell}\right)^{\frac{1}{\varepsilon}} \leq \frac{\delta}{8}.$$

In other words, with probability at most $\delta/8$ no subset $S' \subseteq S$, of cardinality at least $\frac{|S|}{2\beta}$, is a subset of T'. We condition the rest of the proof on the contrary event, denoted by $\mathcal{E}_{T'}$, namely, that there is a subset $S' \subseteq S$ with $S' \subseteq T'$ and $|S'| \geq \frac{|S|}{2\beta}$. Note that S' is then a uniform and independent sample of T' (see Section 2.4.1).

Now, let $c^* \in \mathbb{R}^d_\ell$ be an optimal ℓ-median for T'. The expected distance between $s \in S'$ and c^* is

$$\operatorname{Exp}[\mathrm{d_F}(s, c^*) \mid \mathcal{E}_{T'}] = \sum_{\tau \in T'} \mathrm{d_F}(c^*, \tau) \cdot \frac{1}{|T'|} = \frac{\operatorname{cost}(T', c^*)}{|T'|}.$$

By linearity, we have $\operatorname{Exp}[\operatorname{cost}(S', c^*) \mid \mathcal{E}_{T'}] = \frac{|S'|}{|T'|} \operatorname{cost}(T', c^*)$. Markov's inequality yields:

$$\Pr\left[\frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) > \operatorname{cost}(T', c^*) \;\middle|\; \mathcal{E}_{T'}\right] \leq \frac{\delta}{4}.$$

We conclude that with probability at most $\delta/4$ we have $\frac{\delta|T'|}{4|S'|} \operatorname{cost}(S', c^*) > \operatorname{cost}(T', c^*)$.

Now, from Lemma 3.1.23 we know that there are $2\ell - 4$ subsets $T'_1, \dots, T'_{2\ell-4} \subseteq T'$, of cardinality $\frac{\varepsilon|T'|}{2\ell}$ each and which are not necessarily disjoint, such that for every set $W \subseteq T'$ that contains at least one curve $\tau \in T'_i$ for each $i \in [2\ell - 4]$, there exists a curve $c' \in \mathbb{R}^d_{2\ell-2}$ which has all of its vertices contained in

$$\bigcup_{\tau \in W} \bigcup_{j \in [|\tau|]} B(v^\tau_j, \mathrm{d_F}(\tau, c^*))$$

and for at least $(1 - \varepsilon)|T'|$ curves $\tau \in T' \setminus (T'_1 \cup \dots \cup T'_{2\ell-4})$ it holds that $\mathrm{d_F}(\tau, c') \leq \mathrm{d_F}(\tau, c^*)$.

There are up to $\frac{\varepsilon|T'|}{4\ell}$ curves with distance to c^* at least $\frac{4\ell \operatorname{cost}(T', c^*)}{\varepsilon|T'|}$. Otherwise, the cost of these curves would exceed $\operatorname{cost}(T', c^*)$, which is a contradiction. Later we will prove that each ball we cover has radius at most $\frac{4\ell \operatorname{cost}(T', c^*)}{\varepsilon|T'|}$. Therefore, for each $i \in [2\ell - 4]$ we have to ignore up to half of the curves $\tau \in T'_i$, since we do not cover the balls of radius $\mathrm{d_F}(\tau, c^*)$ centered at their vertices. For each $i \in [2\ell - 4]$ and $s \in S'$ we now have

$$\Pr\left[s \in T'_i \wedge \mathrm{d_F}(s, c^*) \leq \frac{4\ell \operatorname{cost}(T', c^*)}{\varepsilon|T'|} \;\middle|\; \mathcal{E}_{T'}\right] \geq \frac{\varepsilon}{4\ell}.$$

Therefore, by independence, for each $i \in [2\ell-4]$ the probability that no $s \in S'$ is an element of T'_i and has distance to c^* at most $\frac{4\ell \operatorname{cost}(T',c^*)}{\varepsilon|T'|}$ is at most $(1-\frac{\varepsilon}{4\ell})^{|S'|} \leq \exp\left(-\frac{\varepsilon}{4\ell}\frac{4\ell(\ln(4(2\ell-4))-\ln(\delta))}{\varepsilon}\right) = \exp\left(\ln\left(\frac{\delta}{4(2\ell-4)}\right)\right) = \frac{\delta}{4(2\ell-4)}$. Also, with probability at most $\delta/4$ Algorithm 3 fails to compute a 34-approximate ℓ-median $c \in \mathbb{R}^d_\ell$ for S', cf. Corollary 3.1.18.

Using Proposition 2.4.5, we conclude that with probability at least $1-7/8\delta$ all the following events occur simultaneously:

1. There is a subset $S' \subseteq S$ of cardinality at least $|S|/(2\beta)$ that is a uniform and independent sample of T',
2. for each $i \in [2\ell-4]$, S' contains at least one curve from T'_i with distance to c^* up to $\frac{4\ell \operatorname{cost}(T',c^*)}{\varepsilon|T'|}$,
3. Algorithm 3 computes a polygonal curve $c \in \mathbb{R}^d_\ell$ with $\operatorname{cost}(S',c^*_{S'}) \leq \operatorname{cost}(S',c) \leq 34\operatorname{cost}(S',c^*_{S'})$, where $c^*_{S'} \in \mathbb{R}^d_\ell$ is an optimal ℓ-median for S',
4. and it holds that $\frac{\delta|T'|}{4|S'|}\operatorname{cost}(S',c^*) \leq \operatorname{cost}(T',c^*)$.

Let $B_{c^*} = \left\{\tau \in T' \mid \mathrm{d_F}(\tau,c^*) \leq \frac{\operatorname{cost}(T',c^*)}{\varepsilon^2|T'|}\right\}$ and $B_c = \left\{\tau \in T' \mid \mathrm{d_F}(\tau,c) \leq \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}\right\}$. First, note that $|T' \setminus B_{c^*}| \leq \varepsilon^2|T'|$, otherwise $\operatorname{cost}(T' \setminus B_{c^*},c^*) > \operatorname{cost}(T',c^*)$, which is a contradiction, and therefore $|B_{c^*}| \geq (1-\varepsilon^2)|T'|$. We now distinguish two cases:

Case 1: $|B_{c^*} \setminus B_c| > 2\varepsilon|B_{c^*}|$

We have $2\varepsilon|B_{c^*}| \geq (1-\varepsilon^2)2\varepsilon|T'| \geq \varepsilon|T'|$, hence $\Pr\left[\mathrm{d_F}(s,c) > \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} \;\middle|\; \mathcal{E}_{T'}\right] \geq \varepsilon$ for each $s \in S'$. Using independence we conclude that with probability at most

$$(1-\varepsilon)^{|S'|} \leq \exp\left(-\varepsilon\frac{4\ell(\ln(4(2\ell-4))-\ln(\delta))}{\varepsilon}\right) \leq \frac{\delta^{4\ell}}{4^{4\ell}} \leq \frac{\delta}{8}$$

no $s \in S'$ has distance to c greater than $\varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}$. Including this bad event, by Proposition 2.4.5 we conclude that with probability at least $1-\delta$ Items 1 to 4 occur simultaneously and at least one $s \in S'$ has distance to c greater than $\varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}$, hence $\operatorname{cost}(S',c) > \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} \iff \frac{\operatorname{cost}(S',c)}{\varepsilon} > \frac{\operatorname{cost}(T',c^*)}{|T'|}$ and thus we indeed cover the balls of radius at most $\frac{4\ell\operatorname{cost}(T',c^*)}{\varepsilon|T'|} < \frac{4\ell}{\varepsilon}\frac{\operatorname{cost}(S',c^*)}{\varepsilon}$.

In the last step, Algorithm 5 returns a set C of all curves with up to $2\ell-2$ vertices from the grids, that contains one curve, denoted by c'' with same number of vertices as c' (recall that this is the curve guaranteed from Lemma 3.1.23) and distance at most $\frac{\varepsilon}{n}\Delta_l \leq \frac{\varepsilon}{|T'|}\operatorname{cost}(T',c^*)$ between every corresponding pair of vertices of c' and c''. We conclude that $\mathrm{d_F}(c',c'') \leq \frac{\varepsilon}{|T'|}\operatorname{cost}(T',c^*)$. Also, recall that $\mathrm{d_F}(\tau,c') \leq \mathrm{d_F}(\tau,c^*)$ for $\tau \in T' \setminus (T'_1 \cup \dots \cup T'_{2\ell-4})$. Further, T' contains at least $\frac{|T'|}{2}$ curves with distance at most $\frac{2\operatorname{cost}(T',c^*)}{|T'|}$ to c^*, otherwise the cost of the remaining curves would exceed $\operatorname{cost}(T',c^*)$, which is a contradiction, and since $\varepsilon < \frac{1}{2}$ there is at least one curve $\sigma \in T' \setminus (T'_1 \cup \dots \cup T'_{2\ell-4})$ with $\mathrm{d_F}(\sigma,c') \leq \mathrm{d_F}(\sigma,c^*) \leq \frac{2\operatorname{cost}(T',c^*)}{|T'|}$ by the pigeonhole principle. We can now bound the cost of c'' as follows:

$$\begin{aligned}\operatorname{cost}(T',c'') = \sum_{\tau\in T'} \mathrm{d_F}(\tau,c'') &\leq \sum_{\tau\in T'\setminus(T'_1\cup\dots\cup T'_{2\ell-4})} \left(\mathrm{d_F}(\tau,c') + \frac{\varepsilon}{|T'|}\operatorname{cost}(T',c^*)\right) + \\ &\sum_{\tau\in(T'_1\cup\dots\cup T'_{2\ell-4})} (\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,\sigma) + \mathrm{d_F}(\sigma,c') + \mathrm{d_F}(c',c'')) \\ &\leq (1+\varepsilon)\operatorname{cost}(T',c^*) + \sum_{\tau\in(T'_1\cup\dots\cup T'_{2\ell-4})} \left((2+2+\varepsilon)\frac{\operatorname{cost}(T',c^*)}{|T'|}\right) \\ &\leq \operatorname{cost}(T',c^*) + \varepsilon\operatorname{cost}(T',c^*) + 5\varepsilon\operatorname{cost}(T',c^*) = (1+6\varepsilon)\operatorname{cost}(T',c^*).\end{aligned}$$

Case 2: $|B_{c^*}\setminus B_c| \leq 2\varepsilon|B_{c^*}|$

Again, we distinguish two cases:

Case 2.1: $\mathrm{d_F}(c,c^*) \leq 4\varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}$

We can easily bound the cost of c:

$$\operatorname{cost}(T',c) \leq \sum_{\tau\in T'} (\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,c)) \leq (1+4\varepsilon)\operatorname{cost}(T',c^*).$$

Case 2.2: $\mathrm{d_F}(c,c^*) > 4\varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|}$

Recall that $|B_{c^*}| \geq (1-\varepsilon^2)|T'|$. We have

$$\begin{aligned}|T'\setminus B_c| &\leq |T'\setminus B_{c^*}| + 2\varepsilon|B_{c^*}| = |T'| - (1-2\varepsilon)|B_{c^*}| \leq |T'| - (1-2\varepsilon)(1-\varepsilon^2)|T'| \\ &= (2\varepsilon+\varepsilon^2-2\varepsilon^3)|T'| < \frac{1}{3}|T'|.\end{aligned}$$

Hence, $|B_c| \geq (1-2\varepsilon-\varepsilon^2+2\varepsilon^3)|T'| > \frac{2}{3}|T'|$. Assume we assign all curves to c instead of to c^*. For $\tau \in B_c$ we now have decrease in cost $\mathrm{d_F}(\tau,c^*) - \mathrm{d_F}(\tau,c)$, which can be bounded as follows:

$$\begin{aligned}\mathrm{d_F}(\tau,c^*) - \mathrm{d_F}(\tau,c) \geq \mathrm{d_F}(\tau,c^*) - \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} &\geq \mathrm{d_F}(c,c^*) - \mathrm{d_F}(\tau,c) - \varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} \\ &\geq \mathrm{d_F}(c,c^*) - 2\varepsilon\frac{\operatorname{cost}(T',c^*)}{|T'|} > \frac{1}{2}\,\mathrm{d_F}(c,c^*).\end{aligned}$$

For $\tau \in T'\setminus B_c$ we have an increase in cost $\mathrm{d_F}(\tau,c) - \mathrm{d_F}(\tau,c^*) \leq \mathrm{d_F}(c,c^*)$. Let the overall increase in cost be denoted by α, which can be bounded as follows:

$$\alpha < |T'\setminus B_c| \cdot \mathrm{d_F}(c,c^*) - |B_c| \cdot \frac{\mathrm{d_F}(c,c^*)}{2}.$$

By the fact that $|T'\setminus B_c| < \frac{1}{2}|B_c|$ for our choice of ε, we conclude that $\alpha < 0$, which is a contradiction because c^* is an optimal ℓ-median for T'. Therefore, Case 2.2 can not occur. Rescaling ε by $\frac{1}{6}$ proves the claim. □

We analyze the worst-case running time of Algorithm 5 and the number of candidates it returns.

Theorem 3.1.25 *The running time as well as the number of candidates that Algorithm 5 returns is in* $2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}+\log(m)\right)}$.

Proof. The sample S has size $O\left(\frac{\ln(1/\delta)\cdot\beta}{\varepsilon}\right)$ and sampling it takes time $O\left(\frac{\ln(1/\delta)\cdot\beta}{\varepsilon}\right)$. Let $n_S = |S|$. The outer for-loop runs

$$\binom{n_S}{\frac{n_S}{2\beta}} \in 2^{O\left(\frac{n_S}{2\beta}\log n_S\right)} \subset 2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}\right)}$$

times. In each iteration, we run Algorithm 3, taking time $O(m^2\log(m)\ln^2(1/\delta) + m^3\log m)$ (cf. Corollary 3.1.18), we compute the cost of the returned curve with respect to S', taking time $O\left(\frac{\ln(1/\delta)}{\varepsilon}\cdot m\log(m)\right)$, and per curve in S' we build up to m grids of size

$$\left(\frac{\frac{(1+\varepsilon)\Delta}{\varepsilon}}{\frac{2\varepsilon 2\delta n\Delta}{n\sqrt{d}4|S|}}\right)^d = \left(\frac{\sqrt{d}|S|(1+\varepsilon)}{\varepsilon^2\delta}\right)^d \in O\left(\frac{\beta^d(\ln(1/\delta))^d}{\varepsilon^{3d}\delta^d}\right)$$

each. Algorithm 5 then enumerates all combinations of $2\ell-2$ points from up to $|S'|\cdot m$ grids, resulting in

$$O\left(\frac{m^{2\ell-2}\beta^{2\ell d-2d+2\ell-2}(\ln(1/\delta))^{2\ell d-2d+2\ell-2}}{\varepsilon^{6\ell d-6d+2\ell-2}\delta^{2\ell d-2d}}\right)$$

candidates per iteration of the for-loop. Thus, Algorithm 5 computes $O\left(\text{poly}(m,\beta,\delta^{-1},\varepsilon^{-1})\right)$ candidates per iteration of the for-loop and enumeration also takes time $O\left(\text{poly}(m,\beta,\delta^{-1},\varepsilon^{-1})\right)$ per iteration of the for-loop (where $\text{poly}(x_1,x_2,\ldots)$ denotes a polynomial function in $x_1,x_2,\ldots$).

All in all, we have running time and number of candidates $2^{O\left(\frac{\ln^2(1/\delta)\cdot\beta}{\varepsilon^2}+\log(m)\right)}$. □

Since each candidate returned by the algorithm can be evaluated against the input in time $O(nm\log m)$ using Alt and Godau's algorithm, the following corollary follows.

Corollary 3.1.26 *There exists an algorithm that, given a parameter* $\varepsilon \in (0,1)$ *and a set* $T = \{\tau_1,\ldots,\tau_n\} \subset \mathbb{R}^d_m$ *of polygonal curves, returns with constant positive probability a* $(1+\varepsilon)$-*approximate* ℓ-*median with up to* $2\ell-2$ *vertices for* T *in time* $n\cdot 2^{O(\varepsilon^{-2}+\log m)}$.

3.2 Point Sequences

We now study the median problem for point sequences from an arbitrary metric space. We start by formally defining the problem that we study in this section, then we closely review the related work.

3.2.1 Problem Definition

Here, again the most natural way to formulate the problem is as optimization problem of computing a point sequence that minimizes the p-dynamic time warping distance between the given sequences and the median sequence. However, formally we are working with a family of distances, which we incorporate by adding the parameter p to the problem. Furthermore, we extend our definition by incorporating another parameter q to capture related higher order statistics, such as the mean ($q = 2$) [98]. In the literature, the corresponding statistic is commonly named power mean, generalized mean (cf. [67]) or simply q-mean and consequently we name the problem for point sequences (p, q)-mean.

Problem 3.2.1 *The (unrestricted) (p, q)**-mean problem** is defined as follows, where $p, q \in [1, \infty)$ are fixed (constant) parameters of the problem: given a set $T = \{\tau_1, \ldots, \tau_n\} \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ), compute a point sequence $c \in M^*$, such that $\mathrm{cost}_p^q(T, c) = \sum_{i=1}^n \mathrm{d_{DTW}}_p(c, \tau_i)^q$ is minimal.*

As in the ℓ-median problem, we are also interested in computing a mean of bounded (constant) complexity. Another (practical) motivation is to prevent overfitting. We call the corresponding problem the *restricted* (p, q)-mean problem.

Problem 3.2.2 *The **restricted** (p, q)**-mean problem** is defined as follows, where $\ell \in \mathbb{N}_{>1}$ and $p, q \in [1, \infty)$ are fixed (constant) parameters of the problem: given a set $T = \{\tau_1, \ldots, \tau_n\} \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ), compute a point sequence $c \in M^{\leq \ell}$, such that $\mathrm{cost}_p^q(T, c) = \sum_{i=1}^n \mathrm{d_{DTW}}_p(c, \tau_i)^q$ is minimal.*

We note that the case that $p = q$ is prevalent in the literature. Therefore, we call the unrestricted, respectively restricted, (p, p)-mean problem the unrestricted, respectively restricted, p-mean problem. If p and q are clear from the context, we drop them from our notation. Note that for $q = 1$, similar to the previous section, we are dealing with a median problem. However, the representative will be called unrestricted, respectively restricted, (p, q)-mean, or simply mean in this section.

3.2.2 Related Work

Among many practical approaches for the problem of computing a mean, one very influential heuristic is the DTW barycenter averaging (DBA) method, as formalized by Petitjean et al. [215]. The core idea behind DBA is a Lloyd's style (k-means) iterative strategy, which has been rediscovered many times for this problem in the past (see e.g. [220, 1, 140]). DBA iteratively improves the solution as follows: given a candidate average sequence $c = (c_1, \ldots, c_{\ell'})$, it first computes the warpings between c and all input sequences, and then given each set of input vertices S_i matched with the same vertex c_i, it substitutes c_i with the mean of S_i. DBA has inspired many recent solutions that are successful in practice [233, 203, 184, 81, 210]. However, it does not give any guarantees. Just like the k-means algorithm, it may even converge to a local optimum that is arbitrarily far from the global optimum in terms of the target function.

There are few results in the literature with formal guarantees on the running time or the quality of the solution. Brill et al. [49, 50] presented an algorithm for solving the unrestricted 2-mean problem defined over $\mathbb{Q}$ with the Euclidean distance, with an asymptotic bound on the time complexity. Their algorithm is based on dynamic programming and computes the mean in time $O(m^{2n+1}2^n n)$. The algorithm can be slightly modified, to compute a restricted 2-mean. The running time then becomes $O(m^{2n+1}2^n n\ell^2)$. Brill et al. [49, 50] also show that the unrestricted 2-mean problem defined over $\{0, 1\}$ with the Euclidean distance can be solved in $O(nm^3)$ time. This was later improved by Schaar et al. [230] to $O(nm^{1.87})$ time.

All hardness results concern the exact computation of the (p, q)-mean. Bulteau et al. [68] proved that the unrestricted 2-mean problem defined over $\mathbb{Q}$ with the Euclidean distance is NP-hard and W[1]-hard with the number of input sequences n as the parameter. Moreover, they show that the problem cannot be solved in time $O(f(n)) \cdot m^{o(n)}$ for any computable function f unless the Exponential Time Hypothesis[3] fails. Buchin et al. [61] presented an alternative proof of the above statements, which is more general since it applies to the unrestricted (p, q)-mean problem for any $p, q \in \mathbb{N}$. Also, their results hold for the restricted (p, q)-mean problem, when ℓ is part of the input.

We summarize the main algorithmic results on the unrestricted and restricted (p, q)-mean problem.

Problem	Appr.	Running Time	Ambient Space	Reference
r. 2-mean	1	$O(m^{2n+1}2^n n\ell^2)$	$(\mathbb{Q}, (x, y) \mapsto \lVert x - y \rVert)$	[50]
		$(nm)^{2^{O(d\ell)}}$	$(\mathbb{Q}^d, (x, y) \mapsto \lVert x - y \rVert)$	Thm. 3.2.4
u. 2-mean		$O(nm^{1.87})$	$(\{0, 1\}, (x, y) \mapsto \lVert x - y \rVert)$	[230]
		$O(m^{2n+1}2^n n)$		[50]
r. $(p, 1)$-mean	$1 + \varepsilon$	$O(m^4 + nm(\frac{\sqrt[p]{m}}{\varepsilon})^{d\ell} \log m)$	$\mathbb{R}^d$	Thm. 3.2.18
r. p-mean	$2 + \varepsilon$	$O(nm^\ell \varepsilon^{-\ell})$	arbitrary metric space	Thm. 3.2.7
		$O\left(nm(\frac{m}{\varepsilon} \log \frac{m}{\varepsilon})^{\max\{d+1,\ell\}}\right)$	$\mathbb{R}^d$	Thm. 3.2.10
u. 2-mean	∞	$O(nmi)$, i: #iter.	$\mathbb{R}^d$	[215]

We now show that for a special case, namely for point sequences over $(M = \mathbb{Q}^d, \vartheta(x, y) = \lVert x - y \rVert)$, the restricted 2-mean problem can surprisingly be (deterministically) solved exactly.

3.2.3 Exact Computation of a Restricted 2-Mean in Euclidean Space

The idea of our algorithm is to compute for any two warpings between a point sequence of complexity ℓ' and an input point sequence a polynomial function whose sign indicates which of the warpings yields a smaller distance between the sequences. These functions are then used to define an arrangement that partitions the space $\left(\mathbb{R}^d\right)^{\ell'}$. The trick is that while there is an infinite number of point sequences in $\left(\mathbb{R}^d\right)^{\ell'}$, to each input point sequence there are only $O(m^{2\ell'})$ warpings and in each face of the arrangement the point sequences have the same optimal warpings to the input point sequences. Therefore, for an arbitrary point sequence from each face of the arrangement, we can compute the optimal warpings to the input sequences and then use the resulting sections to compute an optimal point sequence for these warpings, obtaining the optimal restricted 2-mean when we eventually hit the face containing it. We use the cylindrical algebraic decomposition algorithm to compute the arrangement and obtain an element of each face.

In the following, we make use of a simplified structure of the solution space, which holds in case $p = q$. This is captured in the notion of sections which we define as follows.

Definition 3.2.3 *Let $T = \{\tau_1, \ldots, \tau_n\} \subseteq M^{\leq m}$ be a set of point sequences and $c = (c_1, \ldots, c_{\ell'}) \in M^{\ell'}$ be a point sequence, both over some metric space (M, ϑ). For $i \in [n]$ and $p \in [1, \infty)$, let W_i be an optimal p-warping between c and τ_i. For $j \in [\ell']$ we define the j^{th} section of c with respect to T (and $W_1, \ldots, W_n$) as follows: $S_j(c, T, W_1, \ldots, W_n) = \{\tau_{i,k} \mid i \in [n], (j, k) \in W_i\}$, where $\tau_{i,k}$ is the k^{th} vertex of τ_i.*

[3] This hypothesis states that there exists a $\delta > 0$ such that 3-SAT (boolean formula satisfiability) can not be solved in time $O(2^{\delta n})$ for formulas of n variables.

If T is clear from the context, we omit it from the notation. Also, we will always omit $W_1, \dots, W_n$ from the notation, because the specific choice of optimal p-warpings is not of interest. We will then write $S_j^p(T, c)$ to clarify that the sections are defined with respect to optimal p-warpings. An immediate consequence of this definition is the following identity:

$$\mathrm{cost}_p^p(T, c) = \sum_{j=1}^{\ell'} \sum_{v \in S_j^p(c,T)} \vartheta(c_j, v)^p,$$

where ℓ' denotes the complexity of c.

A central observation is that the vertices of an optimal restricted p-mean $c = (c_1, \dots, c_{\ell'})$ must minimize the sum of distances, each raised to the p^{th} power, to the vertices in their section, i.e., for all $j \in [\ell']$: $c_j \in \arg\min_{w \in M} \sum_{v \in S_j^p(c,T)} \vartheta(w, v)^p$. Using this, we obtain the following result, which originates from [243].

Theorem 3.2.4 *There exists an algorithm that, given a set $T \subset \left(\mathbb{Q}^d\right)^{\leq m}$ of n point sequences over $(\mathbb{Q}^d, (x, y) \mapsto \|x - y\|)$, computes an optimal restricted 2-mean in time $(nm)^{2^{O(d\ell)}}$.*

Proof. To simplify our exposition, we restrict ourselves to means of complexity exactly $\ell' \in [\ell]$, i.e., in the rest of the proof we describe an algorithm for computing an optimal mean of complexity exactly ℓ'. The complete algorithm consists of iteratively computing the optimal mean of complexity ℓ', for each $\ell' \in [\ell]$.

For each $\tau = (\tau_1, \dots, \tau_{|\tau|}) \in T$ and all $W_1, W_2 \in \mathcal{W}_{\ell',|\tau|}$ we define for $c = (c_1, \dots, c_{\ell'}) \in \left(\mathbb{R}^d\right)^{\ell'}$ the polynomial function

$$P_{\tau,W_1,W_2}(c) = \left(\sum_{(i,j) \in W_1} \|c_i - \tau_j\|^2 \right) - \left(\sum_{(i,j) \in W_2} \|c_i - \tau_j\|^2 \right).$$

Clearly, iff W_1 yields a smaller distance between c and τ than W_2, then $P_{\tau,W_1,W_2}(c) < 0$ and iff W_2 yields a smaller distance between c and τ than W_2, then $P_{\tau,W_1,W_2}(c) > 0$. Iff $P_{\tau,W_1,W_2}(c) = 0$, both yield the same distance.

Let $F = \{P_{\tau,W_1,W_2} \mid \tau \in T, W_1, W_2 \in \mathcal{W}_{\ell',|\tau|}\}$ be the set of these polynomials. The central observation is that if all functions in F have the same sign for any $c_1, c_2 \in \left(\mathbb{R}^d\right)^{\ell'}$, then c_1 and c_2 have the same optimal 2-warpings to the point sequences in T. To see this, for each $\tau \in T$ let $W_\tau \in \mathcal{W}_{\ell',|\tau|}$ be an optimal 2-warping between c_1 and τ. Clearly, $P_{\tau,W_\tau,W}(c_1) \leq 0$ for all $\tau \in T$ and $W \in \mathcal{W}_{\ell',|\tau|}$. Now, if all functions in F have the same sign for c_1 and c_2 it must be that $P_{\tau,W_\tau,W}(c_2) \leq 0$ for all $\tau \in T$ and $W \in \mathcal{W}_{\ell',|\tau|}$. Thus, W_τ is an optimal 2-warping between c_2 and τ for each $\tau \in T$.

Now, we compute an arrangement of the zero sets of the polynomials in F (cf. [195]), i.e., a partition of $\left(\mathbb{R}^d\right)^{\ell'}$ into regions where all functions in F have the same sign. For this purpose we use the cylindrical algebraic decomposition algorithm [76], which also yields a sample from each face of the arrangement and has running time $O(|F|^{f(\ell' \cdot d)})$ for some function $f \in 2^{O(d\ell')}$. For each sample c from some face of the arrangement we first compute the optimal 2-warpings between c and the input point sequences $\tau \in T$ in time $O(nm)$. Second we compute all sections

$S_j(c)$ of c and store the point sequence $c' = (c'_1, \dots, c'_{\ell'})$ consisting of the optimal (cf. e.g. [165]) means $c'_j = \frac{1}{|S_j(c)|} \sum_{v \in S_j(c)} v$, for $j \in [\ell']$. This takes time $O(nm)$.

At some point, we obtain a sample from the face containing the optimal restricted 2-mean $c^* = (c^*_1, \dots, c^*_{\ell'})$ (where c^*_j is the mean of $S_j(c^*)$ for each $j \in [\ell']$), which we return when we finally return the point sequence c' that minimizes the objective function. This takes time $O(nmA)$, where A is the number of cells in the arrangement.

To conclude the proof, note that for each $\tau \in T$ we have that $|\mathcal{W}_{\ell',|\tau|}| \le m^{2\ell'}$, thus $|F| \le nm^{4\ell'}$. Hence, $A \le \left(\frac{100nm^{4\ell'}}{\ell' d}\right)^{\ell' d}$ by [195, Theorem 6.2.1].

As we have already mentioned, we iteratively run the above algorithm to compute means of complexity ℓ', for each $\ell' \in [\ell]$, in order to find an optimal restricted 2-mean. Each iteration runs in $(nm)^{2^{O(d\ell')}} \le (nm)^{2^{O(d\ell)}}$. Since ℓ is constant, the running time is in $(nm)^{2^{O(d\ell)}}$. □

We note that this approach can not be used to compute a restricted p-mean exactly, for any $p \ge 1$. This is due to the fact that a point $x \in \mathbb{R}^d$ that minimizes $\sum_{v \in S_j(c,T)} \|x - v\|^p$ may not be computed exactly for $p \ne 2$ (for $p = 1$ this is the geometric median problem). However, if we have an approximation algorithm A that is able to approximate x, we can modify the approach of Theorem 3.2.4 and let c'_j be the result of A instead of the optimal mean. We obtain the following corollary.

Corollary 3.2.5 *Let $p \in [1, \infty)$ and A be an algorithm that, given a set $P \subset \mathbb{R}^d$, returns a point $x \in \mathbb{R}^d$ with $\min_{y \in \mathbb{R}^d} \sum_{z \in P} \|y - z\|^p \le \sum_{z \in P} \|x - z\|^p \le \alpha \cdot \min_{y \in \mathbb{R}^d} \sum_{z \in P} \|y - z\|^p$, for some $\alpha \ge 1$, in time $T_A(|P|)$. Then there exists an algorithm that, given a set $T \subset \left(\mathbb{Q}^d\right)^{\le m}$ of n point sequences over $(\mathbb{Q}^d, (x,y) \mapsto \|x - y\|)$, computes an α-approximate restricted p-mean in time $T_A(n) \cdot (nm)^{2^{O(d\ell)}}$.*

For $p = 1$, a $(1+\varepsilon)$-approximation algorithm can be achieved by letting A be the algorithm from [75], but the resulting running time is not appealing for an approximation algorithm. Therefore, we now develop more efficient approximation algorithms for the restricted p-mean problem defined on point sequences over an arbitrary metric space and later an improved algorithm for the restricted $(p,1)$-mean problem defined on point sequences over the Euclidean space.

3.2.4 Approximation of a Restricted p-Mean in any Metric Space

We start by describing a simple approximation algorithm that reveals the basic idea underlying the following algorithms. The algorithm relies on the following observation. If p-DTW is defined over a metric space (M, ϑ), then the triangle inequality holds for the point-to-point distances in the sum that defines the p-DTW distance (albeit not for p-DTW distance itself). Assume for simplicity that $p = 1$. In this case, there always exists a 2-approximate median that is formed by points from the input sequences. Enumerating all possible such sequences, then, if the input consists of n point sequences of length m, leads to an algorithm with running time in $O((nm)^{\ell+1})$, where ℓ denotes the upper bound on the complexity of the mean. This approach also extends to other variants of the median problem for different choices of p and q (with varying approximation factors). One obvious disadvantage of this simple algorithm is the high running time.

In the following, we use similar observations as above and show that the dependency on n can be improved to linear while still achieving approximation factors close to 2. We present a randomized constant factor approximation algorithm for the restricted p-mean problem. The approximation factor of the algorithm depends on p, and the best it can achieve is $2+\varepsilon$ for $p=1$ and $4+\varepsilon$ for $p=2$, which resemble the famous Euclidean median and mean problems.

Randomized Algorithm

The idea of the algorithm is to obtain for each $j \in [\ell']$ from the corresponding section $S_j^p(c,T)$ of an optimal restricted p-mean $c^* = (c_1^*,\dots,c_{\ell'}^*)$ one of the closest input vertices to c_j^*. The obtained vertices in the corresponding order form an approximate restricted p-mean. We formalize the idea in the following lemma.

Lemma 3.2.6 *Let $T = \{\tau_1 = (\tau_{1,1},\dots,\tau_{1,|\tau_1|}),\dots,\tau_n = (\tau_{n,1},\dots,\tau_{n,|\tau_n|})\} \subseteq M^{\leq m}$ be a set of point sequences over some metric space (M,ϑ) and let $P = \bigcup_{i=1}^{n}\bigcup_{j=1}^{|\tau_i|}\{\tau_{i,j}\}$. For any $p \in [1,\infty)$, $\ell \in \mathbb{N}_{>1}$ and $\varepsilon \in (0,\infty)$ there exists an $\ell' \leq \ell$ and balls $B_1,\dots,B_{\ell'} \subseteq P$, of cardinality at least $\frac{\varepsilon n}{2^{p-1}+\varepsilon}$ each, such that any point sequence $c = (c_1,\dots,c_{\ell'})$, with $c_i \in B_i$ for each $i \in [\ell']$, is a $(2^p+\varepsilon)$-approximate restricted p-mean for T.*

Proof. Without loss of generality we assume that $\tau_{1,1},\dots,\tau_{1,|\tau_1|},\dots,\tau_{n,1},\dots,\tau_{|\tau_n|}$ are distinct points. Let $c^* = (c_1^*,\dots,c_{\ell'}^*) \in M^{\ell'}$ be an optimal restricted p-mean for T and for $j \in [\ell']$ let $S_j = \{s_{j,1},\dots,s_{j,n_j}\} = S_j^p(c^*)$ for brevity. Define $\Delta(S_j) = \sum\limits_{v \in S_j} \vartheta(c_j^*,v)^p$. We immediately have $\operatorname{cost}(T,c^*) = \sum_{j=1}^{\ell'}\Delta(S_j)$. Now, for $j \in [\ell']$, let $\alpha_j \in \mathcal{S}_{n_j}$ be a permutation of the index set $[n_j]$, such that

$$\vartheta(c_j^*, s_{j,\alpha_j^{-1}(1)})^p \leq \dots \leq \vartheta(c_j^*, s_{j,\alpha_j^{-1}(n_j)})^p.$$

Let $\varepsilon' = \frac{\varepsilon}{2^{p-1}+\varepsilon}$. For the sake of simplicity, we assume that $\varepsilon' n$ is integral. Further, for $j \in [\ell']$, we define $C_j = \{s_{j,\alpha_j^{-1}(1)},\dots,s_{j,\alpha_j^{-1}(\varepsilon' n)}\}$. We have that $\vartheta(c_j^*, s_{j,\alpha_j^{-1}(\varepsilon' n)})^p \leq \frac{\Delta(S_j)}{|S_j|-(\varepsilon' n-1)}$ by the fact that $\vartheta(c_j^*, s_{j,\alpha_j^{-1}(\varepsilon' n)})^p$ is of maximal value, iff $\vartheta(c_j^*,s')^p = 0$ for each $s' \in C_j \setminus \{s_{j,\alpha_j^{-1}(\varepsilon' n)}\}$ and $\vartheta(c_j^*,s')^p = \vartheta(c_j^*, s_{j,\alpha_j^{-1}(\varepsilon' n)})^p$ for each $s' \in S_j \setminus C_j$. For $j \in [\ell']$, we now define

$$B_j = \{x \in P \mid \vartheta(c_j^*,x)^p \leq \vartheta(c_j^*, s_{j,\alpha_j^{-1}(\varepsilon' n)})^p\}$$

and by definition we have $\vartheta(c_j^*,x)^p \leq \frac{\Delta(S_j)}{|S_j|-\varepsilon' n+1} \leq \frac{\Delta(S_j)}{|S_j|-\varepsilon' n}$ for each $x \in B_j$ and $j \in [\ell']$. Then let $c = (c_1,\dots,c_{\ell'})$ be a point sequence with $c_j \in B_j$ for each $j \in [\ell']$. We bound its cost:

$$\begin{aligned}
\operatorname{cost}(T,c) &= \sum_{j=1}^{\ell'}\Delta(S_j) \leq \sum_{j=1}^{\ell'}\sum_{v \in S_j}(\vartheta(c_j^*,v)+\vartheta(c_j^*,c_j))^p \leq \sum_{j=1}^{\ell'}\sum_{v \in S_j} 2^{p-1}(\vartheta(c_j^*,v)^p + \vartheta(c_j^*,c_j)^p) \\
&\leq 2^{p-1}\sum_{j=1}^{\ell'}\sum_{v \in S_j}\left(\vartheta(c_j^*,v)^p + \frac{\Delta(S_j)}{|S_j|-\varepsilon' n}\right) \leq 2^{p-1}\operatorname{cost}(T,c^*) + 2^{p-1}\sum_{j=1}^{\ell'}\sum_{v \in S_j}\frac{\Delta(S_j)}{(1-\varepsilon')|S_j|} \\
&= \left(2^{p-1} + \frac{2^{p-1}}{1-\varepsilon'}\right)\operatorname{cost}(T,c^*) = (2^p+\varepsilon)\operatorname{cost}(T,c^*).
\end{aligned}$$

The first inequality follows from the triangle-inequality and the last inequality holds, because at least one vertex from each $\tau_i \in T$ must be warped to each $c_j^* \in c^*$, thus $|S_j| \geq n$ for each $j \in [\ell']$. □

Now we present the first algorithm for the restricted p-mean problem. The idea is to uniformly sample from the set of all vertices of all point sequences, to obtain at least one vertex from each ball guaranteed by the previous lemma, with high probability. After the sampling, the algorithm enumerates all point sequences of at most ℓ elements from the sample and returns a point sequence with the lowest cost.

Algorithm 6 Restricted p-Mean Constant Factor Approximation (Randomized)

```
procedure MEAN-C($T = \{\tau_1 = (\tau_{1,1}, \dots, \tau_{1,|\tau_1|}), \dots, \tau_n = (\tau_{n,1}, \dots, \tau_{n,|\tau_n|})\}, \delta, \varepsilon, p$)
    $P \leftarrow \bigcup_{i=1}^{n} \bigcup_{j=1}^{|\tau_i|} \{\tau_{i,j}\}$
    $S \leftarrow$ sample $\left\lceil \frac{m(\ln(\ell)+\ln(1/\delta))}{\varepsilon/(2^{p-1}+\varepsilon)} \right\rceil$ points from $P$ uniformly and independently at random
        with replacement
    $C \leftarrow S^{\leq \ell}$
    return an arbitrary element from $\arg\min_{c \in C} \operatorname{cost}_p^p(T, c)$
```

The correctness of Algorithm 6 follows by an application of Lemma 3.2.6.

Theorem 3.2.7 *Given a set $T = \{\tau_1, \dots, \tau_n\} \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ), three parameters $\delta \in (0, 1)$, $\varepsilon \in (0, \infty)$ and $p \in [1, \infty)$, Algorithm 6 returns with probability at least $1 - \delta$ a $(2^p + \varepsilon)$-approximate restricted p-mean for T, in time $O\left(nm^{\ell+1} \ln(1/\delta)^\ell \varepsilon^{-\ell}\right)$.*

Proof. For the given ε, let $\varepsilon' = \frac{\varepsilon}{2^{p-1}+\varepsilon}$ and let $B_1, \dots, B_{\ell'}$, $\ell' \leq \ell$, be the balls guaranteed by Lemma 3.2.6. Recall that each ball has size at least $\varepsilon' n$. For each $i \in [\ell']$ and $s \in S$ we have $\Pr[s \notin B_i] \leq (1 - \frac{\varepsilon' n}{|P|}) \leq (1 - \frac{\varepsilon' n}{nm}) = (1 - \frac{\varepsilon'}{m}) \leq \exp(-\varepsilon'/m)$.

By independence, for each $i \in [\ell']$ we have $\Pr[B_i \cap S = \emptyset] \leq \exp(-\varepsilon'/m)^{\left\lceil \frac{m(\ln(\ell)-\ln(\delta))}{\varepsilon'} \right\rceil} \leq \delta/\ell$. Using a union bound we conclude that with probability at least $1 - \delta$, S contains at least one element of B_i, for each $i \in [\ell']$, and thus Algorithm 6 returns a $(2^p + \varepsilon)$-approximate restricted p-mean for T with probability at least $1 - \delta$ by Lemma 3.2.6.

The running time of the algorithm is dominated by computing the cost of all point sequences of complexity at most ℓ over S. Since $|S^{\leq \ell}|$ is in $O\left(\frac{\ln(1/\delta)^\ell m^\ell}{(\varepsilon')^\ell}\right) \subseteq O\left(\ln(1/\delta)^\ell m^\ell \varepsilon^{-\ell}\right)$ and every distance can be computed in time $O(m)$, this takes time $O\left(\ln(1/\delta)^\ell m^{\ell+1} n \varepsilon^{-\ell}\right)$. □

We now show that this algorithm can be derandomized when M is finite and surprisingly, this derandomization comes at almost no extra cost in the asymptotic running time.

Derandomization

We consider metric spaces (M, ϑ) which together with the set of all (metric) balls $\mathcal{B} = \{\{y \in M \mid \vartheta(x, y) \leq r\} \mid x \in M, r \in \mathbb{R}_{\geq 0}\}$ form a range space $(M, \mathcal{B})$ with bounded VC dimension $\mathcal{D}$. We present a deterministic algorithm for the restricted p-mean problem which is applicable under the additional assumption that there is a subsystem oracle for $(P, \mathcal{B}_{|P})$ (where $P \subseteq M$ is finite), which is the case for the Euclidean metric. Note that interestingly Huang et al. [146] show that if one allows $(1 \pm \varepsilon)$ distortion on the original distances of a metric space with bounded doubling dimension, then the VC dimension of the range space induced by the metric balls is also bounded as a function of the doubling dimension and ε. However, our algorithm also depends on the existence of a subsystem oracle which is not always obvious for a given metric.

The following algorithm is a modification of Algorithm 6 where the sampling step is substituted for a computation of an ε-net of the set of all vertices of all given point sequences. Since the balls guaranteed by Lemma 3.2.6 are of appropriate size, the ε-net intersects all of them and by enumeration of all point sequences of at most ℓ points from the ε-net, we again find a good approximate restricted p-mean.

Algorithm 7 Restricted p-Mean Constant Factor Approximation (Deterministic)

```
procedure MEAN-C-D($T = \{\tau_1 = (\tau_{1,1}, \dots, \tau_{1,|\tau_1|}), \dots, \tau_n = (\tau_{n,1}, \dots, \tau_{n,|\tau_n|})\}, \varepsilon, p$)
    $\varepsilon' \leftarrow \frac{\varepsilon}{2^{p-1}+\varepsilon}$, $P \leftarrow \bigcup_{i=1}^{n} \bigcup_{j=1}^{|\tau_i|} \{\tau_{i,j}\}$
    $S \leftarrow$ compute an $(\varepsilon'/m)$-net of $(P, \mathcal{B}_{|P})$
    $C \leftarrow S^{\leq \ell}$
    return an arbitrary element from $\arg\min_{c \in C} \mathrm{cost}_p^p(T, c)$
```

The correctness of Algorithm 7 follows from Definition 2.6.5 and Lemma 3.2.6.

Theorem 3.2.8 *Given a set $T \subseteq M^{\leq m}$ of n point sequences over some metric space (M, ϑ) and parameters $\varepsilon \in (0, \infty)$ and $p \in [1, \infty)$, Algorithm 7 returns a $(2^p + \varepsilon)$-approximate restricted p-mean for T.*

Proof. By Lemma 3.2.6, for any $\varepsilon \in (0, \infty)$ there exist balls $B_1, \dots, B_{\ell'} \subseteq P$, $\ell' \leq \ell$, of cardinality at least $\varepsilon' n$ each, such that any point sequence $c = (c_1, \dots, c_{\ell'})$, with $c_i \in B_i$ for each $i \in [\ell']$, is a $(2^p + \varepsilon)$-approximate restricted p-mean for T, where $\varepsilon' = \frac{\varepsilon}{2^{p-1}+\varepsilon}$. Since we compute an (ε'/m)-net of P and $|P| \leq nm$, S contains at least one point from each of $B_1, \dots, B_{\ell'}$ by Definition 2.6.5. Hence, $S^{\leq \ell}$ contains a $(2^p + \varepsilon)$-approximate restricted p-mean for T. □

We now turn to the Euclidean setting; formally, we now have $M = \mathbb{R}^d$ and $\vartheta(x, y) = \|x - y\|$. First, we prove that there exist a subsystem oracle for $(P, \mathcal{B}_{|P})$, when $P \subseteq M$ is a finite subset of the d-dimensional Euclidean space.

Lemma 3.2.9 *There is a subsystem oracle for the range space $(P, \mathcal{B}_{|P})$, where P is a finite subset of $\mathbb{R}^d$.*

Proof. The VC dimension of $(P, \mathcal{B}_{|P})$ is bounded by $d + 1$, see [130]. For any $Y \subseteq P$, we need to compute the set $\mathcal{B}_{|Y}$ explicitly in time $O(|Y|^{d+2})$. We first apply the standard lifting (cf. [195]) $\phi \colon (x_1, \dots, x_d) \mapsto \left(x_1, \dots, x_d, \sum_{i=1}^{d} x_i^2\right)$. A point $p \in Y$ belongs to some ball $B \in \mathcal{B}$, with center $c = (c_1, \dots, c_d) \in \mathbb{R}^d$ and radius $r > 0$, if and only if, $\phi(p)$ lies below the hyperplane h_B, where h_B is the hyperplane defined by the equation $\langle a_B, x \rangle = b_B$, where $a_B = (2c_1, 2c_2, \dots 2c_d, 1)$ and $b_B = r^2 - \sum_{i=1}^{d} c_i^2$. Notice that h_B is non-vertical by definition. Then we dualize: for any point $\phi(p) = (y_1, \dots, y_{d+1})$, $D(\phi(p)) = \{(x_1, \dots, x_{d+1}) \in \mathbb{R}^{d+1} \mid x_{d+1} = \sum_{i=1}^{d} x_i y_i - y_{d+1}\}$ is a non-vertical hyperplane in $\mathbb{R}^{d+1}$ and for any non-vertical hyperplane h_B, $D^{-1}(h_B)$ is a point in $\mathbb{R}^{d+1}$. A standard fact about duality (cf. [195]) is that a point $\phi(p)$ lies below a hyperplane h_B if and only if the hyperplane $D(\phi(p))$ lies above point $D^{-1}(h_B)$. Finally we construct the arrangement of hyperplanes in the dual space in time $O(|Y|^{d+1})$, using the algorithm in [91]. For each of the at most $O(|Y|^{d+1})$ cells, we return a subset $X \subseteq Y$ corresponding to the hyperplanes lying above. The overall running time is $O(|Y|^{d+2})$. □

Finally, we can analyze the running time of Algorithm 7 in the Euclidean setting.

Theorem 3.2.10 *Given a set $T \subset \left(\mathbb{R}^d\right)^{\leq m}$ of n point sequences and parameters $\varepsilon \in (0,\infty)$ and $p \in [1,\infty)$, Algorithm 7 can be implemented to run in time*[4] $O\left(nm\left(\left(\frac{m}{\varepsilon}\log\frac{m}{\varepsilon}\right)^{d+1} + \left(\frac{m}{\varepsilon}\log\frac{m}{\varepsilon}\right)^{\ell}\right)\right)$.

Proof. The VC dimension of the range space $(P, \mathcal{B}_{|P})$ is bounded by $d+1$, see [130]. By Lemma 3.2.9, we can use Theorem 2.6.7 to compute an (ε'/m)-net S of $(P, \mathcal{B}_{|P})$, with size $|S| \in O\left(\frac{m}{\varepsilon}\log\left(\frac{m}{\varepsilon}\right)\right)$, in time $O\left(nm\left(\frac{m}{\varepsilon}\log\left(\frac{m}{\varepsilon}\right)\right)^{d+1}\right)$. We then compute the $\mathrm{d}_{\mathrm{DTW}_p}$ distance of any of the candidates from $|S|^{\leq \ell}$ with the n input point sequences in time $O\left(\ell|S|^{\ell}\cdot nm\right)$. □

3.2.5 $(1+\varepsilon)$-Approximation of a Restricted $(p,1)$-Mean in Euclidean Space

Here, we study the restricted $(p,1)$-mean problem. This is exactly the problem of computing one median point sequence of complexity at most ℓ, under the p-DTW distance.

First, we introduce an idea that is central in the main result, which is a randomized approximation algorithm. We formalize this idea in the following theorem, which uses the weak triangle inequality and provides an upper bound on the expected cost of the restricted $(p,1)$-mean obtained by first sampling an input point sequence uniformly at random and then computing an α-approximate minimum-error ℓ-simplification of this point sequence.

Theorem 3.2.11 *Let $T = \{\tau_1, \dots, \tau_n\} \subseteq M^{\leq m}$ be a set of point sequences over some metric space (M, ϑ) and let $p \in [1,\infty)$. Let σ be a point sequence sampled uniformly at random from T, and let σ' be an α-approximate minimum-error ℓ-simplification of σ under $\mathrm{d}_{\mathrm{DTW}_p}$, where $\ell \leq m$. Then,*

$$\mathrm{Exp}\left[\mathrm{cost}_p^1(T,\sigma')\right] \leq (2+\alpha)(m\ell)^{\frac{1}{p}}\,\mathrm{cost}_p^1(T,c^*),$$

where c^* is an optimal restricted $(p,1)$-mean of T.

Proof. We have

$$\begin{aligned}
\mathrm{Exp}\left[\mathrm{cost}_p^1(T,\sigma')\right] &= \mathrm{Exp}\left[\sum_{i=1}^n \mathrm{d}_{\mathrm{DTW}_p}(\tau_i,\sigma')\right] \leq \mathrm{Exp}\left[m^{\frac{1}{p}}\sum_{i=1}^n \left(\mathrm{d}_{\mathrm{DTW}_p}(\tau_i,c^*) + \mathrm{d}_{\mathrm{DTW}_p}(c^*,\sigma')\right)\right] && \text{(I)}\\
&= m^{\frac{1}{p}}\cdot\left(\mathrm{cost}_p^1(T,c^*) + n\cdot\mathrm{Exp}[\mathrm{d}_{\mathrm{DTW}_p}(c^*,\sigma')]\right)\\
&\leq m^{\frac{1}{p}}\cdot\left(\mathrm{cost}_p^1(T,c^*) + n\ell^{\frac{1}{p}}\cdot\mathrm{Exp}[\mathrm{d}_{\mathrm{DTW}_p}(c^*,\sigma) + \mathrm{d}_{\mathrm{DTW}_p}(\sigma,\sigma')]\right) && \text{(II)}\\
&\leq m^{\frac{1}{p}}\cdot\left(\mathrm{cost}_p^1(T,c^*) + (1+\alpha)n\ell^{\frac{1}{p}}\cdot\mathrm{Exp}[\mathrm{d}_{\mathrm{DTW}_p}(c^*,\sigma)]\right)\\
&= m^{\frac{1}{p}}\cdot\left(\mathrm{cost}_p^1(T,c^*) + (1+\alpha)n\ell^{\frac{1}{p}}\cdot\sum_{\tau\in T}\mathrm{d}_{\mathrm{DTW}_p}(c^*,\tau)\cdot\frac{1}{n}\right)\\
&= m^{\frac{1}{p}}\cdot\left(\mathrm{cost}_p^1(T,c^*) + (1+\alpha)\ell^{\frac{1}{p}}\cdot\mathrm{cost}_p^1(T,c^*)\right)\\
&\leq (2+\alpha)(m\ell)^{\frac{1}{p}}\cdot\mathrm{cost}_p^1(T,c^*),
\end{aligned}$$

where in Eq. (I) and in Eq. (II) we applied Lemma 2.3.6. □

[4] We assume d to be constant.

We again turn to point sequences in the Euclidean space. Recall that we are formally dealing with point sequences over $(M = \mathbb{R}^d, \vartheta(x, y) = \|x - y\|)$ and we are interested in computing a median point sequence $c \in \left(\mathbb{R}^d\right)^{\leq \ell}$ under p-DTW. In the following, we design a $(1 + \varepsilon)$-approximation algorithm for the corresponding restricted $(p, 1)$-mean problem. The algorithm is randomized and succeeds with probability at least $1 - \delta$, where $\delta \in (0, 1)$ is a user-defined parameter.

The high-level idea is the following. Given a set T of n point sequences, we first compute a rough estimate of the optimal cost. To do so, we sample a sufficiently large number of input sequences that we store in a set S, and we compute a 2-approximate minimum-error ℓ-simplification for each one of them. We detect a sequence in S whose simplification minimizes the restricted $(p, 1)$-mean cost, denoted by R. Since S is of appropriate size, a combination of Theorem 3.2.11 with Markov's inequality implies that with good probability, R is a $4(m\ell)^{\frac{1}{p}}$-approximation of the optimal cost.

We can now use R to "guess" a refined estimate for the restricted $(p, 1)$-mean cost which is within a constant factor from the optimal cost, by enumerating multiples of 2 in the interval $[R/8 \cdot (m\ell)^{-\frac{1}{p}}, R]$. Assuming that we have such an estimate, we can use it to fine-tune a grid that covers balls (of suitable radii) centered at the points of sequences in S. We use the resulting grid points to compute a set of candidate solutions. The idea here is that with good probability one of the point sequences in S is very close to the optimal solution, so one of the candidate solutions will be a good approximation.

Algorithm 8 Restricted $(p, 1)$-Mean $(1 + \varepsilon)$-Approximation

```
procedure MED-APPR($T = \{\tau_1 = (\tau_{1,1}, \dots, \tau_{1,|\tau_1|}), \dots, \tau_n = (\tau_{n,1}, \dots, \tau_{n,|\tau_n|})\}, \varepsilon, p, \delta$)
    $S \leftarrow$ sample $\lceil \log(2/\delta) \rceil$ point sequences from $T$ uniformly and independently
          at random with replacement
    $\mathcal{R} \leftarrow \emptyset$, $C \leftarrow \emptyset$
    for each $\tau_i \in S$ do
        $\tau_i' \leftarrow$ 2-approximate minimum-error $\ell$-simplification of $\tau_i$ under $\mathrm{d}_{\mathrm{DTW}_p}$ (Algorithm 2)
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{\mathrm{cost}_p^1(T, \tau_i')\}$
    $R \leftarrow \min \mathcal{R}$
    $\beta \leftarrow 2 \cdot \left(\frac{68 m^{1/p}}{\varepsilon} + 5\right)^d$
    $I_R \leftarrow \left\{\frac{R \cdot 2^{-i}}{n} \mid i \in \{0\} \cup [\lceil 3 + \log(m\ell)/p \rceil]\right\}$
    for each $r \in I_R$ do
        $\gamma \leftarrow \frac{\varepsilon \cdot r}{(2m)^{1/p}\sqrt{d}}$
        for each $\tau_i \in S$ do
            $\mathcal{B}(\tau_i, 4r) \leftarrow \bigcup_{j=1}^{|\tau_i|} B(\tau_{i,j}, 4r)$
            $N \leftarrow \mathbb{G}(\mathcal{B}(\tau_i, 4r), \gamma)$
            if $|N| \leq \ell \cdot \beta$ then
                $C \leftarrow C \cup N^{\leq \ell}$
    return an arbitrary element from $\arg\min_{c \in C} \mathrm{cost}_p^1(T, c)$.
```

Now we analyze the running time and correctness of Algorithm 8. We begin with a bound on the probability that R is a rough approximation of the optimal $(p, 1)$-mean cost.

Lemma 3.2.12 *Let c^* be an optimal restricted $(p, 1)$-mean of T. With probability at least $1 - \delta/2$ it holds that $R \leq 8(m\ell)^{\frac{1}{p}} \mathrm{cost}(T, c^*)$.*

Proof. For any $\tau \in T$, let τ' be a 2-approximate minimum-error ℓ-simplification of τ. Now, let τ be a point sequence sampled uniformly at random from T. By Theorem 3.2.11 it holds that

$$\mathrm{Exp}[\mathrm{cost}(T, \tau')] \leq 4(m\ell)^{\frac{1}{p}} \cdot \mathrm{cost}(T, c^*).$$

By Markov's inequality we have that $\Pr\left[\mathrm{cost}(T, \tau') \geq 8(m\ell)^{\frac{1}{p}} \cdot \mathrm{cost}(T, c^*)\right] \leq \frac{1}{2}$. Hence, the probability that $R \geq 8(m\ell)^{\frac{1}{p}} \cdot \mathrm{cost}(T, c^*)$ is

$$\Pr\left[\min_{\tau \in S} \mathrm{cost}(T, \tau') \geq 8(m\ell)^{1/p} \cdot \mathrm{cost}(T, c^*)\right] \leq \frac{1}{2^{|S|}} \leq \frac{\delta}{2}.$$

□

Next, we bound the probability that a point sequence in the sample S is conveniently close to the optimal $(p, 1)$-mean.

Lemma 3.2.13 *Let c^* be an optimal restricted $(p, 1)$-mean of T. With probability at least $1 - \delta/2$, there exists a $\tau \in S$ such that $\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \leq \frac{2}{n} \cdot \mathrm{cost}(T, c^*)$.*

Proof. Let τ be a point sequence sampled uniformly at random from T. We have

$$\mathrm{Exp}[\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*)] = \sum_{i=1}^{n} \mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \cdot \frac{1}{n} = \frac{\mathrm{cost}(T, c^*)}{n}.$$

By Markov's inequality it holds that $\Pr\left[\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \geq 2 \cdot \frac{\mathrm{cost}(T,c^*)}{n}\right] \leq \frac{1}{2}$. Hence,

$$\Pr\left[\min_{\tau \in S} \mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \geq 2 \cdot \frac{\mathrm{cost}(T, c^*)}{n}\right] \leq \frac{1}{2^{|S|}} \leq \frac{\delta}{2}.$$

□

The set I_R contains a value r such that nr is within a factor of 2 from the optimal cost.

Lemma 3.2.14 *Let c^* be an optimal restricted $(p, 1)$-mean of T. If $R \leq 8(m\ell)^{\frac{1}{p}} \mathrm{cost}(T, c^*)$, then there exists an $r \in I_R$ such that $\mathrm{cost}(T, c^*) \in [nr, 2nr]$.*

Proof. We have that $\mathrm{cost}(T, c^*) \leq R$, since R is the cost of an ℓ-simplification, and by assumption it holds that $\mathrm{cost}(T, c^*) \geq \frac{R}{8}(m\ell)^{-\frac{1}{p}}$. By the definition of I_R, there exists a $j \in [\lceil 3+\log(m\ell)/p \rceil] \cup \{0\}$ such that

$$2^{-(j+1)} \cdot \frac{R}{n} \leq \frac{\mathrm{cost}(T, c^*)}{n} \leq 2^{-j} \cdot \frac{R}{n}.$$

Hence, the lemma is true for $r = 2^{-(j+1)} \cdot \frac{R}{n}$. □

The following is an upper bound on the number of grid cells needed to cover a Euclidean ball. Similar bounds often appear in the literature, but they are typically asymptotic and not sufficient for our needs. Therefore, we prove an exact (non-asymptotic) upper bound.

Lemma 3.2.15 *Let $x \in \mathbb{R}^d$ and $r, \gamma \in (0, \infty)$. It holds that*

$$|\mathbb{G}\left(B(x, 8r), \gamma\right)| \leq 2 \cdot \left(\frac{34r}{\gamma\sqrt{d}} + 5\right)^d.$$

Proof. We use Binet's second expression [253] for the natural logarithm of the Gamma function:

$$\ln \Gamma(z) = z \ln(z) - z + \frac{1}{2} \ln\left(\frac{2\pi}{z}\right) + \int_0^\infty \frac{2 \arctan\left(\frac{t}{z}\right)}{e^{2\pi t} - 1} \mathrm{d}t.$$

Since $\arctan(x) \geq 0$ for $x \geq 0$ and $e^{2\pi x} - 1 \geq 0$ for $x \geq 0$, we have the following inequality:

$$\begin{aligned} \ln \Gamma(z) \geq z \ln(z) - z + \frac{1}{2} \ln\left(\frac{2\pi}{z}\right) &\iff \ln \Gamma(z) \geq \ln(z^z) - \ln(e^z) + \ln\left(\sqrt{\frac{2\pi}{z}}\right) \\ &\iff \Gamma(z) \geq z^z e^{-z} \sqrt{\frac{2\pi}{z}} \\ &\iff \Gamma(z) \geq \sqrt{2\pi} z^{z - \frac{1}{2}} e^{-z}. \qquad \text{(I)} \end{aligned}$$

We apply a standard volumetric argument to upper bound $|\mathbb{G}\left(B(x, 8r), \gamma\right)|$, where $\mathrm{vol}(P)$ denotes the d-dimensional volume (Lebesgue measure) of any $P \subseteq \mathbb{R}^d$.

$$\begin{aligned} |\mathbb{G}\left(B(x, 8r), \gamma\right)| &\leq \frac{\mathrm{vol}(B(x, 8r + \gamma\sqrt{d}))}{\gamma^d} = \frac{\pi^{d/2}}{\Gamma(\frac{d}{2} + 1)} \cdot \frac{(8r + \gamma\sqrt{d})^d}{\gamma^d} \\ &\leq \frac{\pi^{d/2} e^{d/2+1}}{\sqrt{2\pi}\left(\frac{d}{2} + 1\right)^{d/2+1/2}} \cdot \frac{(8r + \gamma\sqrt{d})^d}{\gamma^d} \qquad \text{(II)} \\ &\leq \frac{2^{d/2+1/2} \pi^{d/2} e^{d/2+1}}{\sqrt{2\pi} \cdot d^{d/2+1/2}} \cdot \frac{(8r + \gamma\sqrt{d})^d}{\gamma^d} \\ &\leq \frac{e \cdot (4.2)^d}{\sqrt{\pi} \cdot d^{d/2}} \cdot \frac{(8r + \gamma\sqrt{d})^d}{\gamma^d} \leq 2 \cdot \left(\frac{34r}{\gamma\sqrt{d}} + 5\right)^d, \end{aligned}$$

where in the first equation we use the formula for the volume of a d-dimensional ball from [195] and also that the volume of a cell from a grid of width γ is γ^d. Furthermore, in Eq. (II) we use Eq. (I). □

We now focus on the iteration of the algorithm with $r \in I_R$, $\tau_i \in S$, such that r satisfies the property guaranteed by Lemma 3.2.14 and τ_i satisfies the property guaranteed by Lemma 3.2.13. We claim that in this certain iteration, an $(1+\varepsilon)$-approximate restricted $(p, 1)$-mean is added to the set of candidates C.

Lemma 3.2.16 *Let* $c^* = (c_1^*, \ldots, c_{\ell'}^*)$ *be an optimal restricted* $(p, 1)$*-mean of* T*. Let* $r^* \in I_r$ *be chosen such that* $\mathrm{cost}(T, c^*) \in [nr^*, 2nr^*]$ *and let* $\gamma^* = \frac{\varepsilon r^*}{(2m)^{1/p}\sqrt{d}}$*. Let* $\tau = (\tau_1, \ldots, \tau_{|\tau|}) \in S$ *and assume that* $\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \leq \frac{2}{n} \cdot \mathrm{cost}(T, c^*)$ *then*

1. $|\mathbb{G}(\mathcal{B}(\tau, 4r^*), \gamma^*)| \leq \ell \cdot 2 \cdot \left(\frac{34r^*}{\gamma^*\sqrt{d}} + 5\right)^d$ *and*
2. *there exists a* $c \in \mathbb{G}(\mathcal{B}(\tau, 4r^*), \gamma^*)^{\leq \ell}$ *such that* $\mathrm{cost}(T, c) \leq (1 + \varepsilon) \cdot \mathrm{cost}(T, c^*)$*.*

Proof. To prove Item 1, notice that $\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \leq 4r^*$, which implies that for any vertex τ_i of τ, there exists a vertex c_j^* of c^* such that $\tau_i \in B(c_j^*, 4r^*)$. By the triangle inequality $B(\tau_i, 4r^*) \subseteq B(c_j^*, 8r^*)$. Hence,

$$\mathcal{B}(\tau, 4r^*) \subseteq \bigcup_{j=1}^{\ell'} B(c_j^*, 8r^*) \implies |\mathbb{G}(\mathcal{B}(\tau, 4r^*), \gamma^*)| \leq \left|\mathbb{G}\left(\bigcup_{j=1}^{\ell'} B(c_j^*, 8r^*), \gamma^*\right)\right| \leq \sum_{j=1}^{\ell'} \left|\mathbb{G}\left(B(c_j^*, 8r^*), \gamma^*\right)\right|.$$

By Lemma 3.2.15, we obtain

$$|\mathbb{G}(\mathcal{B}(\tau, 4r^*), \gamma^*)| \leq \ell \cdot 2 \cdot \left(\frac{34r^*}{\gamma^* \sqrt{d}} + 5\right)^d.$$

To prove Item 2, notice that all vertices of c^* are contained in $\mathcal{B}(\tau, 4r^*)$. Hence, for each vertex c_j^* there exists a grid point $c_j \in \mathbb{G}(\mathcal{B}(\tau, 4r^*), \gamma^*)$ such that $\|c_j^* - c_j\| \leq \gamma^* \sqrt{d}$. We will show that the point sequence $c = (c_1, \ldots, c_{\ell'})$ is a $(1+\varepsilon)$-approximate restricted $(p,1)$-mean for T. For each $i \in [n]$, let W_i^* denote an optimal warping between τ_i and c^*.

$$\begin{aligned}
\operatorname{cost}(T, c) &= \sum_{i=1}^n \mathrm{d}_{\mathrm{DTW}_p}(\tau_i, c) = \sum_{i=1}^n \min_{W \in \mathcal{W}_{|\tau_i|, \ell'}} \left(\sum_{(j,k) \in W} \|\tau_{i,j} - c_k\|^p\right)^{\frac{1}{p}} \\
&\leq \sum_{i=1}^n \left(\sum_{(j,k) \in W_i^*} \|\tau_{i,j} - c_k\|^p\right)^{\frac{1}{p}} \leq \sum_{i=1}^n \left(\sum_{(j,k) \in W_i^*} (\|\tau_{i,j} - c_k^*\| + \|c_k^* - c_k\|)^p\right)^{\frac{1}{p}} \\
&\leq \sum_{i=1}^n \left(\left(\sum_{(j,k) \in W_i^*} \|\tau_{i,j} - c_k^*\|^p\right)^{\frac{1}{p}} + \left(\sum_{(j,k) \in W_i^*} \|c_k^* - c_k\|^p\right)^{\frac{1}{p}}\right) \\
&\leq \sum_{i=1}^n \left(\mathrm{d}_{\mathrm{DTW}_p}(\tau_i, c^*) + |W_i^*|^{1/p} \cdot \gamma^* \sqrt{d}\right) \leq \sum_{i=1}^n \left(\mathrm{d}_{\mathrm{DTW}_p}(\tau_i, c^*) + \frac{\operatorname{cost}(T, c^*) \cdot \varepsilon}{n}\right) \\
&= (1+\varepsilon) \cdot \operatorname{cost}(T, c^*),
\end{aligned}$$

where the second inequality follows from the triangle inequality, and the third inequality follows from Minkowski's inequality [226]. We also make use of the fact that $|W_i^*| \leq 2m$. □

We now prove the correctness of Algorithm 8.

Lemma 3.2.17 *Given a finite set $T \subset \left(\mathbb{R}^d\right)^{\leq m}$ of point sequences, three parameters $\delta \in (0,1)$, $\varepsilon \in (0, \infty)$ and $p \in [1, \infty)$, Algorithm 8 returns with probability at least $1-\delta$ a $(1+\varepsilon)$-approximate restricted $(p,1)$-mean for T.*

Proof. Let c^* be an optimal restricted $(p,1)$-mean of T. Applying a union bound over the events of Lemma 3.2.12 and Lemma 3.2.13, we conclude that with probability at least $1-\delta$ it holds that $R \leq 8(m\ell)^{\frac{1}{p}} \cdot \operatorname{cost}(T, c^*)$ and there exists a $\tau \in S$ such that $\mathrm{d}_{\mathrm{DTW}_p}(\tau, c^*) \leq (2/n) \cdot \operatorname{cost}(T, c^*)$. We show correctness assuming that the above two events hold. By Lemma 3.2.14 we know that there exists an $r^* \in I_R$ such that $\operatorname{cost}(T, c^*) \in [nr^*, 2nr^*]$.

We focus on the iteration where r^* is considered. Let γ^* be the value of γ in that iteration and let N^* be the set N in that iteration. By Lemma 3.2.16 Item 1, $|N^*| \leq \ell\beta$ and all point sequences of complexity at most ℓ defined by points in N^* will be considered as possible solutions. Finally, by Lemma 3.2.16 Item 2, there is a point sequence in $(N^*)^{\leq \ell}$ which is a $(1+\varepsilon)$-approximate solution. □

Finally, we bound the running time of Algorithm 8.

Theorem 3.2.18 *Given a set $T \subset \left(\mathbb{R}^d\right)^{\leq m}$ of n point sequences, three parameters $\delta \in (0,1)$, $\varepsilon \in (0, m^{1/p}]$ and $p \in [1,\infty)$, Algorithm 8 returns a $(1+\varepsilon)$-approximate restricted $(p,1)$-mean with probability at least $1-\delta$ and has running time $O\left(\left(m^4 + nm \cdot \left(\frac{m^{1/p}}{\varepsilon}\right)^{d\ell} \cdot \log(m)\right) \cdot \log(1/\delta)\right)$.*

Proof. The correctness follows from Lemma 3.2.17. It remains to bound the running time.

For each one of the point sequences in S, we compute its $(2,\ell)$-simplification in $O(dm^4\ell)$ time using Algorithm 2 (see Theorem 2.8.4) and its $(p,1)$-mean cost in $O(dnm\ell)$ time. Hence, the total time needed to compute $\mathcal{R}$ and then R is $O((m^4+nm)\cdot d\ell\log(1/\delta))$. The set I_R has cardinality $|I_R| \in O\left(\frac{\log(m\ell)}{p}\right)$. For each value $r \in I_R$, we add at most $|N|^{\leq \ell} \cdot |S| = \ell\left(\ell \cdot 2 \cdot \left(\frac{68m^{1/p}}{\varepsilon}+5\right)^d\right)^\ell \cdot |S|$ candidates. For each candidate point sequence in C, we compute the cost in time $O(dnm\ell)$. Since d, p and ℓ are considered constants, the total running time is $O\left(\left(m^4 + nm \cdot \left(\frac{m^{1/p}}{\varepsilon}\right)^{d\ell} \cdot \log(m)\right) \cdot \log(1/\delta)\right)$. □

4 k-Median Clustering

Clustering is the task of unveiling a hidden structure in a given set of objects. More specifically, we want to partition the set into so-called clusters such that the elements are more similar within a cluster than across clusters. The nature of the objects as well as the measure of (dis-)similarity depends on the application at hand and indeed this diverse and versatile concept has to this day been studied under numerous expressions, most prominently in the fields of data mining and machine learning, with applications such as summarization, learning, segmentation, and target marketing. It has probably first been investigated about six decades ago and was influenced by a range of scientific disciplines. Today it subsumes a plethora of methods, techniques and applications, which can not be covered by this manuscript. For an overview on the matter we refer the reader to one of the detailed books and surveys on the topic, e.g. [255, 13, 153, 118, 155, 154].

In this work, we focus on a particular manifestation of the concept: center based clustering, for an introduction see e.g. [130, Chapter 4]. Here, each cluster is induced by its representative center, i.e., it consists of the elements that are closer to its center than to any another center. Arguably the most popular problems in center based clustering are *Euclidean k-center*, *k-median* and *k-means*. Here, one is given a set of points in $\mathbb{R}^d$ and wants to compute k center points from $\mathbb{R}^d$ that minimize the maximum Euclidean distance between a given point and its nearest center point, respectively the sum of Euclidean distances between the given points and their nearest center point, respectively the sum of squared Euclidean distances between the given points and their nearest center. These problems are naturally motivated by applications like data summarization [170], facility location [212] and learning a mixture of Gaussians [189].

The k-center and k-median problems are also popular in a more general setting, where input and cluster centers come from an arbitrary finite metric space. However, metric k-center is NP-hard to approximate within a factor of $(2 - \varepsilon)$ [123] and metric k-median is NP-hard to approximate within a factor of $((1 + 2/e) - \varepsilon)$ [156]. Their Euclidean counterparts are also hard to solve and some are even hard to approximate. In general, Euclidean k-center, k-median and k-means are NP-hard [130]. Euclidean k-means is even NP-hard if either k or d is fixed [19, 192] and Euclidean k-median is NP-hard when d is fixed [197]. Euclidean k-center and k-means are NP-hard to approximate within a factor of $((1 + \sqrt{7})/2 - \varepsilon)$ [100], respectively within a factor of $(1.0013 - \varepsilon)$ [178]. However, if k, and particularly d, are fixed, then there exist polynomial time approximation schemes for Euclidean k-median and k-means [132]. We further note that k-median clustering has also been considered in a non-metric setting for distance measures like Bregman divergences [6].

Popular clustering algorithms that are used in practice are the greedy 2-approximation algorithm for k-center clustering by Gonzalez [123], a local search heuristic for k-median clustering by Arya et al. [25] and the famous heuristic by Lloyd [185] for k-means clustering.

Of course, here we are interested in spatial data sequences and due to the versatility of the concept, clustering of sequential data such as time series, trajectories, and texts, particularly DNA, has been widely studied to this day. Again, the topic is too broad to provide a general overview, so we refer the reader to one of the numerous surveys on the matter, e.g. [181, 36, 14, 158].

In this work, we focus on a notion of clustering that has only recently been introduced [87] and has quickly become popular [59, 207, 60, 44, 61]. We name it (k,ℓ)-clustering and it subsumes the (k,ℓ)-center and (k,ℓ)-median objectives, which are derived from k-center and k-median. Here, one is given a set of polygonal curves/point sequences of complexity at most m each and wants to compute k polygonal curves/point sequences of complexity at most ℓ each, such that the maximum distance between an input curve and a nearest center curve, respectively sum of distances between the input curves and their nearest center curve, is minimized. These objectives incorporate the complexity restriction on the centers (recall the introduction of Chapter 3) and are commonly equipped with variants of the Fréchet and the dynamic time warping distances.

In this chapter we build upon the results obtained in Chapter 3 and we study variants of (k,ℓ)-median clustering under the (continuous) Fréchet and (discrete) dynamic time warping distance. First, we modify an existing algorithm to approximate a generalized k-median clustering problem, which captures (k,ℓ)-median clustering. Furthermore, we develop an algorithm that constructs ε-coresets for this generalized k-median clustering problem under the restriction that the input comes from a metric space. We formally define the problem, such that it is suited to capture the different data types and measures that we use.

4.1 Problem Definition

In the following, let $\mathcal{X} = (X, \rho)$ be an arbitrary space, where X is any non-empty (ground-)set and $\rho\colon X \times X \to \mathbb{R}_{\geq 0}$ is a distance function (not necessarily a metric). We introduce a generalized definition of k-median clustering, where the input is restricted to come from a predefined subset $Y \subseteq X$ and the medians are restricted to come from a predefined subset $Z \subseteq X$.

Problem 4.1.1 *The **generalized k-median clustering problem** is defined as follows, where $k \in \mathbb{N}$ is a fixed (constant) parameter of the problem: given a finite and non-empty set $T = \{\tau_1, \dots, \tau_n\} \subseteq Y$, compute a set C of k elements from Z, such that $\operatorname{cost}(T, C) = \sum_{\tau \in T} \min_{c \in C} \rho(\tau, c)$ is minimal.*

Later in this work, Y will be the domain of the high-complexity input and Z will be the domain of the low-complexity (approximate) medians. However, this abstract definition is of more general interest.

4.2 Algorithm for Clustering in Metric and Non-Metric Spaces

Here we present an algorithm that is able to approximate Problem 4.1.1. Before we do so, we review the related work.

4.2.1 Related Work

Our algorithm builds upon the clustering algorithm by Ackermann et al. [6], which is a generalization of the algorithm by Kumar et al. [173]. The algorithm by Kumar et al. was developed to approximate a class of Euclidean clustering problems, including Euclidean k-median and k-means, as well as discrete k-means clustering. It achieves with high probability an approximation factor of $(1+\varepsilon)$ and has running time $nd2^{(k/\varepsilon)^{O(1)}}$. In this version of the algorithm, sampling, particularly superset sampling (recall Section 2.4.1), is hardwired into the candidate phase. In this vain, it is shown that any clustering problem that satisfies a certain sampling property can be approximated by the algorithm. Finally, Kumar et al. provide an extension to weighted k-median and k-means clustering. The algorithms require integral weights and achieve a $(1+\varepsilon)$-approximation with high probability in time $nd2^{(k/\varepsilon)^{O(1)}} \log^k W$, where W is the sum of all weights.

The generalized algorithm by Ackermann et al. was developed to perform k-median clustering using non-metric distance measures. Their research originated in a project that demanded to identify a set of representatives in a large set of probability distributions, whose distances are measured using the Kullback-Leibler divergence. In the course, they noted that almost no approximation algorithms were known for non-metric clustering problems, a discrepancy to the popularity of non-metric distance measures used in practice. Ackermann et al. [6] present a randomized $(1+\varepsilon)$-approximation algorithm for k-median clustering of probability distributions under the Kullback-Leibler divergence, of speech data under the Itakura-Saito divergence, of points in $\mathbb{R}^d$ under the Mahalanobis distance and some special and related cases of Bregman divergences, and also of points from certain metric spaces with bounded doubling dimension. They achieve this by a simplified and purely combinatorial analysis of the algorithm by Kumar et al., which does not require metric properties. Also, they generalize the sampling property developed by Kumar et al.. The running time of their algorithm is roughly $n2^{O(k \log k/\varepsilon)}$.

4.2.2 The Algorithm

The algorithm in [6] is a recursive approximation scheme that employs two phases in each call. In the so-called *candidate phase* it computes candidates by taking a sample S from the input set T and running an algorithm on each subset of S of a certain size. Which algorithm to use, depends on the metric at hand. The idea behind this is simple: if T contains a cluster T' that takes a constant fraction of its size, then a constant fraction of S is from T' with high probability. By brute-force enumeration of all subsets of S, one can find this subset $S' \subseteq T'$ and if S is taken uniformly and independently at random from T then S' is a uniform and independent sample from T' (see Section 2.4.1). Ackermann et al. proved for various metric and non-metric distance measures that S' can be used for computing candidates that contain a $(1+\varepsilon)$-approximate median for T' with high probability. The algorithm recursively calls itself for each candidate to eventually evaluate these together with the candidates for the remaining clusters.

The second phase of the algorithm is the so-called *pruning phase*, where it partitions its input according to the candidates at hand into two sets of equal size: one with the smaller distances to the candidates and one with the larger distances to the candidates. It then recursively calls itself with the second set as input. The idea behind this is that small clusters become larger and will eventually take a constant fraction of the input, such that candidates for these can be found in the candidate phase. Furthermore, the partitioning yields a provably small error. Finally, when k centers are found in each call, the algorithm returns the set of k candidates that together evaluated best against the input.

We generalize the algorithm in the following way: instead of drawing a uniform sample and running a problem-specific algorithm on this sample in the candidate phase, we only run a problem-specific "plugin"-algorithm in the candidate phase, thus dropping the framework around the sampling property. We think that the problem-specific algorithms used in [6] do not fulfill the role of a plugin, since parts of the problem-specific operations, e.g. the uniform sampling, remain in the main algorithm. Here, we separate the problem-specific operations from the main algorithm: any algorithm can serve as plugin, if it is able to return candidates for a cluster that takes a constant fraction of the input, where the fraction is an input-parameter of the algorithm and some approximation factor is guaranteed (with high probability). The calls to the candidate-finder plugin do not even need to be (stochastically) independent, allowing adaptive algorithms.

The following algorithm, Algorithm 9, can approximate every k-median problem compatible with Problem 4.1.1, when provided with a problem-specific plugin-algorithm for computing candidates. It has several parameters. The first parameter C is the set of centers yet found and κ is the number of centers yet to be found. The following parameters concern only the plugin-algorithm used within the algorithm: β determines the size (in terms of a fraction of the input) of the smallest cluster for which an approximate median can be computed, δ determines the probability of failure of the plugin-algorithm, and ε determines the approximation factor of the plugin-algorithm.

Algorithm 9 Recursive Approximation-Scheme for k-Median Clustering

```
procedure $k$-MEDIAN$(T, C, \kappa, \beta, \delta, \varepsilon)$
    if $\kappa = 0$ then
        return $C$
                                                        ▷ Pruning Phase
    if $C \neq \emptyset$ and $|T| > 1$ then
        $P \leftarrow$ set of $\left\lfloor \frac{|T|}{2} \right\rfloor$ elements $\tau \in T$, such that $\min_{c \in C} \rho(\tau, c) \leq \min_{c \in C} \rho(\sigma, c)$ for each $\sigma \in T \setminus P$
        $C' \leftarrow k$-MEDIAN$(T \setminus P, C, \min\{\kappa, |T \setminus P|\}, \beta, \delta, \varepsilon)$
    else
        $C' \leftarrow \emptyset$
                                                        ▷ Candidate Phase
    $K \leftarrow$ MEDIAN-CANDIDATES$(T, \beta, \delta/k, \varepsilon)$
    for $c \in K$ do
        $C_c \leftarrow k$-MEDIAN$(T, C \cup \{c\}, \min\{\kappa - 1, |T|\}, \beta, \delta, \varepsilon)$
    $\mathcal{C} \leftarrow \{C'\} \cup \bigcup_{c \in K} \{C_c\}$
    return $\arg\min_{C \in \mathcal{C}} \operatorname{cost}(T, C)$
```

The quality of approximation and worst case running time of Algorithm 9 is stated in the following two theorems, which we prove further below. The proofs are adaptations of corresponding proofs in [6]. We provide them for the sake of completeness. We note that no metric properties are used in the proofs.

Theorem 4.2.1 *Let $\alpha \in [1, \infty)$ and* MEDIAN-CANDIDATES *be an algorithm that, given three parameters $\beta \in [1, \infty)$, $\delta, \varepsilon \in [0, 1)$ and a finite set $T \subseteq Y$, returns with probability at least $1 - \delta$ an $(\alpha + \varepsilon)$-approximate 1-median for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$.*

Let $T \subseteq Y$ be a finite set. Algorithm 9 called with parameters $(T, \emptyset, k, \beta, \delta, \varepsilon)$, where $\beta \in (2k, \infty)$ and $\delta, \varepsilon \in (0, 1)$, returns with probability at least $1 - \delta$ a set $C = \{c_1, \dots, c_k\}$ with $\operatorname{cost}(T, C) \leq (1 + \frac{4k^2}{\beta - 2k})(\alpha + \varepsilon) \operatorname{cost}(T, C^*)$, where C^* is an optimal set of k medians for T.

Theorem 4.2.2 *Let $T_1(n, \beta, \delta, \varepsilon)$ denote the worst case running time of* MEDIAN-CANDIDATES *for an arbitrary input set T with $|T| = n$ and let $C(n, \beta, \delta, \varepsilon)$ denote the maximum number of candidates it returns. Also, let T_ρ denote the worst case running time needed to compute ρ for an input element and a candidate.*

If T_1 and C are non-decreasing in n, Algorithm 9 has running time $O(C(n, \beta, \delta, \varepsilon)^{k+2} \cdot n \cdot T_\rho + C(n, \beta, \delta, \varepsilon)^{k+1} \cdot T_1(n, \beta, \delta, \varepsilon))$.

The following proof is an adaption of [6, Theorem 2.2 - Theorem 2.5].

Proof of Theorem 4.2.1. For $k = 1$, the claim trivially holds. We now distinguish two cases. In the first case the principle of the proof is presented in all its detail. In the second case we only show how to generalize the first case to $k > 2$.

Case 1: $k = 2$

Let $C^* = \{c_1^*, c_2^*\}$ be an optimal set of k medians for T with clusters T_1^* and T_2^*, respectively, that form a partition of T. For the sake of simplicity, assume that n is a power of 2 and w.l.o.g. assume that $|T_1^*| \geq \frac{1}{2}|T| > \frac{1}{\beta}|T|$. Let C_1 be the set of candidates returned by MEDIAN-CANDIDATES in the initial call. With probability at least $1 - \delta/k$, there is a $c_1 \in C_1$ with $\operatorname{cost}(T_1^*, c_1) \leq (\alpha + \varepsilon) \operatorname{cost}(T_1^*, c_1^*)$. We distinguish two cases:

Case 1.1: There exists a recursive call with parameters $(T', \{c_1\}, 1, \beta, \delta, \varepsilon)$ and $|T_2^* \cap T'| \geq \frac{1}{\beta}|T'|$.

First, we assume that T' is the maximum cardinality input with $|T_2^* \cap T'| \geq \frac{1}{\beta}|T'|$, occurring in a recursive call of the algorithm. Let C_2 be the set of candidates returned by MEDIAN-CANDIDATES in this call. With probability at least $1 - \delta/k$, there is a $c_2 \in C_2$ with $\operatorname{cost}(T_2^* \cap T', c_2) \leq (\alpha + \varepsilon) \operatorname{cost}(T_2^* \cap T', \widetilde{c}_2)$, where $\widetilde{c}_2$ is an optimal median for $T_2^* \cap T'$.

Let P be the set of elements of T removed in the $m \in \mathbb{N}$, $m \leq \log(n)$, pruning phases between obtaining c_1 and c_2. Without loss of generality we assume that $P \neq \emptyset$. For $i \in [m]$, let P_i be the elements removed in the i^{th} (in the order of the recursive calls occurring) pruning phase. Note that the P_i are pairwise disjoint, we have that $P = \cup_{i=1}^{t} P_i$ and $|P_i| = \frac{n}{2^i}$. Since $T = T_1^* \uplus (T_2^* \cap T') \uplus (T_2^* \cap P)$, we have

$$\operatorname{cost}(T, \{c_1, c_2\}) \leq \operatorname{cost}(T_1^*, c_1) + \operatorname{cost}(T_2^* \cap T', c_2) + \operatorname{cost}(T_2^* \cap P, c_1). \tag{I}$$

Our aim is now to prove that the number of elements wrongly assigned to c_1, i.e., $T_2^* \cap P$, is small and further, that their cost is a fraction of the cost of the elements correctly assigned to c_1, i.e., T_1^*.

We define $R_0 = T$ and for $i \in [m]$ we define $R_i = R_{i-1} \setminus P_i$. The R_i are the elements remaining after the i^{th} pruning phase. Note that by definition $|R_i| = \frac{n}{2^i} = |P_i|$. Since $R_m = T'$ is the maximum cardinality input, with $|T_2^* \cap T'| \geq \frac{1}{\beta}|T'|$, we have that $|T_2^* \cap R_i| < \frac{1}{\beta}|R_i|$ for all $i \in [m-1]$. Also, for each $i \in [m]$ we have $P_i \subseteq R_{i-1}$, therefore

$$|T_2^* \cap P_i| \leq |T_2^* \cap R_{i-1}| < \frac{1}{\beta}|R_{i-1}| = \frac{2}{\beta}\frac{n}{2^i} \tag{II}$$

and as immediate consequence

$$|T_1^* \cap P_i| = |P_i| - |T_2^* \cap P_i| > |P_i| - \frac{1}{\beta}|R_{i-1}| = \left(1 - \frac{2}{\beta}\right)\frac{n}{2^i}. \tag{III}$$

This tells us that mainly the elements of T_1^* are removed in the pruning phase and only very few elements of T_2^*. By definition, we have for all $i \in [m-1]$, $\sigma \in P_i$ and $\tau \in P_{i+1}$ that $\rho(\sigma, c_1) \leq \rho(\tau, c_1)$, hence

$$\frac{1}{|T_2^* \cap P_i|} \operatorname{cost}(T_2^* \cap P_i, c_1) \leq \frac{1}{|T_1^* \cap P_{i+1}|} \operatorname{cost}(T_1^* \cap P_{i+1}, c_1).$$

Combining this inequality with Eqs. (II) and (III) we obtain for $i \in [m-1]$:

$$\begin{aligned} &\frac{\beta 2^i}{2n} \operatorname{cost}(T_2^* \cap P_i, c_1) < \frac{2^{i+1}}{(1-2/\beta)n} \operatorname{cost}(T_1^* \cap P_{i+1}, c_1) \\ \iff &\operatorname{cost}(T_2^* \cap P_i, c_1) < \frac{2^{i+1} 2n}{(1-2/\beta)n\beta 2^i} \operatorname{cost}(T_1^* \cap P_{i+1}, c_1) = \frac{4}{(\beta - 2)} \operatorname{cost}(T_1^* \cap P_{i+1}, c_1). \quad \text{(IV)} \end{aligned}$$

We still need such a bound for $i = m$. Since $|R_m| = |P_m|$ and also $R_m \subseteq R_{m-1}$ we can use Eq. (II) to obtain:

$$|T_1^* \cap R_m| = |R_m| - |T_2^* \cap R_m| \geq |R_m| - |T_2^* \cap R_{m-1}| > \left(1 - \frac{2}{\beta}\right) \frac{n}{2^m}. \quad \text{(V)}$$

Also, we have for all $\sigma \in P_m$ and $\tau \in R_m$ that $\rho(\sigma, c_1) \leq \rho(\tau, c_1)$ by definition, thus

$$\frac{1}{|T_2^* \cap P_m|} \operatorname{cost}(T_2^* \cap P_m, c_1) \leq \frac{1}{|T_1^* \cap R_m|} \operatorname{cost}(T_1^* \cap R_m, c_1).$$

We combine this inequality with Eq. (II) and Eq. (V) and obtain:

$$\begin{aligned} &\frac{\beta 2^m}{2n} \operatorname{cost}(T_2^* \cap P_m, c_1) < \frac{2^m 2n}{(1-2/\beta)n\beta 2^m} \operatorname{cost}(T_1^* \cap R_m, c_1) \\ \iff &\operatorname{cost}(T_2^* \cap P_m, c_1) < \frac{2}{(\beta - 2)} \operatorname{cost}(T_1^* \cap R_m, c_1). \quad \text{(VI)} \end{aligned}$$

We are now ready to bound the cost of the elements of T_2^* wrongly assigned to c_1. Combining Eq. (IV) and Eq. (VI) yields:

$$\begin{aligned} \operatorname{cost}(T_2^* \cap P, c_1) &= \sum_{i=1}^{m} \operatorname{cost}(T_2^* \cap P_i, c_1) < \frac{4}{\beta - 2} \sum_{i=1}^{m-1} \operatorname{cost}(T_1^* \cap P_{i+1}, c_1) + \frac{2}{\beta - 2} \operatorname{cost}(T_1^* \cap R_m, c_1) \\ &< \frac{4}{\beta - 2} \operatorname{cost}(T_1^*, c_1). \end{aligned}$$

Here, the last inequality holds, because $P_2, \ldots, P_m$ and R_m are pairwise disjoint. Also, we have

$$\operatorname{cost}(T_2^* \cap T', c_2) \leq (\alpha + \varepsilon) \operatorname{cost}(T_2^* \cap T', \widetilde{c}_2) \leq (\alpha + \varepsilon) \operatorname{cost}(T_2^* \cap T', c_2^*) \leq (\alpha + \varepsilon) \operatorname{cost}(T_2^*, c_2^*).$$

Finally, using Eq. (I) and a union bound, with probability at least $1 - \delta$ the following holds:

$$\begin{aligned} \operatorname{cost}(T, \{c_1, c_2\}) &< (\alpha + \varepsilon) \operatorname{cost}(T_1^*, c_1^*) + (\alpha + \varepsilon) \operatorname{cost}(T_2^*, c_2^*) + \frac{4}{\beta - 2} (\alpha + \varepsilon) \operatorname{cost}(T_1^*, c_1^*) \\ &< \left(1 + \frac{4}{\beta - 2}\right) (\alpha + \varepsilon) \operatorname{cost}(T, C^*) = \left(1 + \frac{4k}{k\beta - 2k}\right) (\alpha + \varepsilon) \operatorname{cost}(T, C^*) \\ &\leq \left(1 + \frac{4k^2}{\beta - 2k}\right) (\alpha + \varepsilon) \operatorname{cost}(T, C^*). \end{aligned}$$

Case 1.2: For all recursive calls with parameters $(T', \{c_1\}, 1, \beta, \delta, \varepsilon)$ it holds that $|T_2^* \cap T'| < \frac{1}{\beta} |T'|$.

After $\log(n)$ pruning phases we end up with a singleton $\{\sigma\} = T'$ as input set. Since $|T_2^* \cap T'| < \frac{1}{\beta} |T'|$, it must be that $0 = |T_2^* \cap T'| < \frac{1}{\beta} |T'| = \frac{1}{\beta} < 1$ and thus $\sigma \in T_1^*$.

Let C_2 be the set of candidates returned by Median-Candidates in this call. With probability at least $1 - \delta/k$ there is a $c_2 \in C_2$ with $\operatorname{cost}(\{\sigma\}, c_2) \leq (\alpha + \varepsilon) \operatorname{cost}(\{\sigma\}, \widetilde{c}_2) \leq$

$(\alpha+\varepsilon)\operatorname{cost}(\{\sigma\}, c_1^*)$, where $\widetilde{c}_2$ is an optimal median for $\{\sigma\}$. Since $\operatorname{cost}(T_2^* \cap P, c_1)$ is bounded as in Case 1.1, by a union bound we have with probability at least $1-\delta$:

$$\begin{aligned}\operatorname{cost}(T, \{c_1, c_2\}) &\leq \operatorname{cost}(T_1^* \setminus \{\sigma\}, c_1) + \operatorname{cost}(T_2^* \cap P, c_1) + \operatorname{cost}(\{\sigma\}, c_2)\\ &\leq (\alpha+\varepsilon)\operatorname{cost}(T_1^*, c_1^*) + \operatorname{cost}(T_2^* \cap P, c_1)\\ &\leq \left(1 + \frac{4}{\beta-2}\right)(\alpha+\varepsilon)\operatorname{cost}(T, C^*)\\ &\leq \left(1 + \frac{4k^2}{\beta-2k}\right)(\alpha+\varepsilon)\operatorname{cost}(T, C^*).\end{aligned}$$

Case 2: $k > 2$

We only prove the generalization of Case 1.1 to $k > 2$, the remainder of the proof is analogous to the Case 1. Let $C^* = \{c_1^*, \ldots, c_k^*\}$ be an optimal set of k medians for T with clusters $T_1^*, \ldots, T_k^*$, respectively, that form a partition of T. For the sake of simplicity, assume that n is a power of 2 and w.l.o.g. assume $|T_1^*| \geq \cdots \geq |T_k^*|$. For $i \in [k]$ and $j \in [k] \setminus [i]$ we define $T_{i,j}^* = \uplus_{t=i}^{j} T_t^*$.

Let $\mathcal{T}_0 = T$ and let $(\mathcal{T}_j = \mathcal{T}_{j-1} \setminus \mathcal{P}_j)_{j=1}^m$ be the sequence of input sets in the recursive calls of the $m \in \mathbb{N}$, $m \leq \log(n)$, pruning phases, where $\mathcal{P}_j$ is the set of elements removed in the j^{th} (in the order of the recursive calls occurring) pruning phase. Let $\mathcal{T} = \{\mathcal{T}_0\} \cup \{\mathcal{T}_j \mid j \in [m]\}$. For $i \in [k]$, let T_i be the maximum cardinality set in $\mathcal{T}$, with $|T_i^* \cap T_i| \geq \frac{1}{\beta}|T_i|$. Note that by assumption and since $\beta > 2k$, $T_1 = T$ must hold and also $T_j \subset T_i$ for $j \in [k] \setminus [i]$.

Using a union bound, with probability at least $1-\delta$, for each $i \in [k]$ the call of Median-Candidates with input T_i yields a candidate c_i with

$$\operatorname{cost}(T_i^* \cap T_i, c_i) \leq (\alpha+\varepsilon)\operatorname{cost}(T_i^* \cap T_i, \widetilde{c}_i) \leq (\alpha+\varepsilon)\operatorname{cost}(T_i^* \cap T_i, c_i^*) \leq (\alpha+\varepsilon)\operatorname{cost}(T_i^*, c_i^*), \tag{I}$$

where $\widetilde{c}_i$ is an optimal 1-median for $T_i^* \cap T_i$. Let $C = \{c_1, \ldots, c_k\}$ be the set of these candidates and for $i \in [k-1]$, let $P_i = T_i \setminus T_{i+1}$ denote the set of elements of T removed by the pruning phases between obtaining c_i and c_{i+1}. Note that the P_i are pairwise disjoint.

By definition, the sets

$$T_1^* \cap T_1, \ldots, T_k^* \cap T_k, T_{2,k}^* \cap P_1, \ldots, T_{k,k}^* \cap P_{k-1}$$

form a partition of T, therefore

$$\begin{aligned}\operatorname{cost}(T, \{c_1, \ldots, c_k\}) &\leq \sum_{i=1}^{k}\operatorname{cost}(T_i^* \cap T_i, c_i) + \sum_{i=1}^{k-1}\operatorname{cost}\Big(T_{i+1,k}^* \cap P_i, \{c_1, \ldots, c_i\}\Big)\\ &\leq (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T_i^*, c_i^*) + \sum_{i=1}^{k-1}\operatorname{cost}\Big(T_{i+1,k}^* \cap P_i, \{c_1, \ldots, c_i\}\Big). \qquad \text{(II)}\end{aligned}$$

Now, it only remains to bound the cost of the wrongly assigned elements of $T_{i+1,k}^*$. For $i \in [k]$, let $n_i = |T_i|$ and w.l.o.g. assume that $P_i \neq \emptyset$ for each $i \in [k-1]$. Each P_i is the disjoint union $\uplus_{j=1}^{m_i} P_{i,j}$ of $m_i \in \mathbb{N}$ sets of elements of T removed in the interim pruning phases, and it holds that $|P_{i,j}| = \frac{n_i}{2^j}$. We now prove for each $i \in [k-1]$ and $j \in [m_i]$ that P_i contains many elements from $T_{1,i}^*$ and only a few elements from $T_{i+1,k}^*$.

For $i \in [k-1]$, we define $R_{i,0} = T_i$ and for $j \in [m_i]$ we define $R_{i,j} = R_{i,j-1} \setminus P_{i,j}$. By definition, $|R_{i,j}| = \frac{n_i}{2^j} = |P_{i,j}|$, $R_{i,j_1} \supset R_{i,j_2}$ for each $j_1 \in [m_i]$ and $j_2 \in [m_i] \setminus [j_1]$, also $R_{i,m_i} = T_{i+1}$. Thus,

$|T_t^* \cap R_{i,j}| < \frac{1}{\beta}|R_{i,j}|$ for all $i \in [k-1], j \in [m_i]$ and $t \in [k] \setminus [i]$. As immediate consequence we obtain $|T_{i+1,k}^* \cap R_{i,j}| \leq \frac{k}{\beta}|R_{i,j}|$. Since $P_{i,j} \subseteq R_{i,j-1}$ for all $i \in [k-1]$ and $j \in [m_i]$, we have

$$|T_{i+1,k} \cap P_{i,j}| \leq |T_{i+1,k} \cap R_{i,j-1}| \leq \frac{k}{\beta}|R_{i,j-1}| = \frac{2k}{\beta}\frac{n_i}{2^j}, \tag{III}$$

which immediately yields

$$|T_{1,i} \cap P_{i,j}| = |P_{i,j}| - |T_{i+1,k} \cap P_{i,j}| \geq \left(1 - \frac{2k}{\beta}\right)\frac{n_i}{2^j}. \tag{IV}$$

Now, by definition we know that for all $i \in [k-1]$, $j \in [m_i] \setminus \{m_i\}$, $\sigma \in P_{i,j}$ and $\tau \in P_{i,j+1}$ that $\min\limits_{c \in \{c_1,\dots,c_i\}} \rho(\sigma, c) \leq \min\limits_{c \in \{c_1,\dots,c_i\}} \rho(\tau, c)$. Thus,

$$\frac{\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,j}, \{c_1,\dots,c_i\}\right)}{|T_{i+1,k}^* \cap P_{i,j}|} \leq \frac{\operatorname{cost}\left(T_{1,i}^* \cap P_{i,j+1}, \{c_1,\dots,c_i\}\right)}{|T_{1,i}^* \cap P_{i,j+1}|}.$$

Combining this inequality with Eqs. (III) and (IV) yields for $i \in [k-1]$ and $j \in [m_i] \setminus \{m_i\}$:

$$\begin{aligned} &\frac{\beta 2^j}{2kn_i}\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,j}, \{c_1,\dots,c_i\}\right) \leq \frac{2^{j+1}}{(1-\frac{2k}{\beta})n_i}\operatorname{cost}\left(T_{1,i}^* \cap P_{i,j+1}, \{c_1,\dots,c_i\}\right) \\ \iff &\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,j}, \{c_1,\dots,c_i\}\right) \leq \frac{4k}{\beta - 2k}\operatorname{cost}\left(T_{1,i}^* \cap P_{i,j+1}, \{c_1,\dots,c_i\}\right) \end{aligned} \tag{V}$$

For each $i \in [k-1]$ we still need an upper bound on $\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,m_i}, \{c_1,\dots,c_i\}\right)$. Since $|R_{i,m_i}| = |P_{i,m_i}|$ and also $R_{i,m_i} \subseteq R_{i,m_i-1}$ we can use Eq. (III) to obtain

$$|T_{1,i}^* \cap R_{i,m_i}| = |R_{i,m_i}| - |T_{i+1,k}^* \cap R_{i,m_i}| \geq |R_{i,m_i}| - |T_{i+1,k}^* \cap R_{i,m_i-1}| > \left(1 - \frac{2k}{\beta}\right)\frac{n_i}{2^{m_i}}. \tag{VI}$$

By definition, we also know that for all $i \in [k-1]$, $\sigma \in P_{i,m_i}$ and $\tau \in R_{i,m_i}$ that $\min\limits_{c \in \{c_1,\dots,c_i\}} \rho(\sigma, c) \leq \min\limits_{c \in \{c_1,\dots,c_i\}} \rho(\tau, c)$. Thus,

$$\frac{\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,m_i}, \{c_1,\dots,c_i\}\right)}{|T_{i+1,k}^* \cap P_{i,m_i}|} \leq \frac{\operatorname{cost}\left(T_{1,i}^* \cap R_{i,m_i}, \{c_1,\dots,c_i\}\right)}{|T_{1,i}^* \cap R_{i,m_i}|}.$$

Combining this inequality with Eqs. (III) and (VI) yields:

$$\begin{aligned} &\frac{\beta 2^{m_i}}{2kn_i}\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,m_i}, \{c_1,\dots,c_i\}\right) < \frac{2^{m_i}}{(1-\frac{2k}{\beta})n_i}\operatorname{cost}\left(T_{1,i}^* \cap R_{i,m_i}, \{c_1,\dots,c_i\}\right) \\ \iff &\operatorname{cost}\left(T_{i+1,k}^* \cap P_{i,m_i}, \{c_1,\dots,c_i\}\right) < \frac{2k}{\beta - 2k}\operatorname{cost}\left(T_{1,i}^* \cap R_{i,m_i}, \{c_1,\dots,c_i\}\right). \end{aligned} \tag{VII}$$

We can now give the following bound, combining Eqs. (V) and (VII), for each $i \in [k-1]$:

$$\begin{aligned}
\operatorname{cost}\Big(T^*_{i+1,k} \cap P_i, \{c_1,\ldots,c_i\}\Big) &= \sum_{j=1}^{m_i} \operatorname{cost}\Big(T^*_{i+1,k} \cap P_{i,j}, \{c_1,\ldots,c_i\}\Big) \\
&< \sum_{j=1}^{m_i-1} \frac{4k}{\beta-2k} \operatorname{cost}\Big(T^*_{1,i} \cap P_{i,j+1}, \{c_1,\ldots,c_i\}\Big) \\
&\quad + \frac{2k}{\beta-2k} \operatorname{cost}\Big(T^*_{1,i} \cap R_{i,m_i}, \{c_1,\ldots,c_i\}\Big) \\
&< \frac{4k}{\beta-2k} \operatorname{cost}\Big(T^*_{1,i} \cap T_i, \{c_1,\ldots,c_i\}\Big). \qquad \text{(VIII)}
\end{aligned}$$

Here, the last inequality holds, because $P_{i,2},\ldots,P_{i,m_i}$ and R_{i,m_i} are pairwise disjoint subsets of T_i.

Now, we plug this bound into Eq. (II). Note that $T^*_j \cap T_i \subseteq T^*_j \cap T_j$ for each $i \in [k]$ and $j \in [i]$ by definition. We obtain:

$$\begin{aligned}
\operatorname{cost}(T,\{c_1,\ldots,c_k\}) &\leq (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) + \sum_{i=1}^{k-1}\operatorname{cost}\Big(T^*_{i+1,k}\cap P_i,\{c_1,\ldots,c_i\}\Big) \\
&< (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) + \frac{4k}{\beta-2k}\sum_{i=1}^{k-1}\operatorname{cost}\Big(T^*_{1,i}\cap T_i,\{c_1,\ldots,c_i\}\Big) \\
&\leq (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) + \frac{4k}{\beta-2k}\sum_{i=1}^{k-1}\sum_{t=1}^{i}\operatorname{cost}(T^*_t\cap T_i,c_t) \\
&\leq (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) + \frac{4k}{\beta-2k}\sum_{i=1}^{k-1}\sum_{t=1}^{i}\operatorname{cost}(T^*_t\cap T_t,c_t) \\
&\leq (\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) + \frac{4k^2}{\beta-2k}\sum_{i=1}^{k-1}\operatorname{cost}(T^*_i\cap T_i,c_i) \\
&\leq \left(1+\frac{4k^2}{\beta-2k}\right)(\alpha+\varepsilon)\sum_{i=1}^{k}\operatorname{cost}(T^*_i,c^*_i) = \left(1+\frac{4k^2}{\beta-2k}\right)(\alpha+\varepsilon)\operatorname{cost}(T,C^*).
\end{aligned}$$

The last inequality follows from Eq. (I). □

The following analysis of the worst case running time of Algorithm 5 is a slight adaption of [6, Theorem 2.8], which is also provided for the sake of completeness.

Proof of Theorem 4.2.2. Let $T(n,\kappa,\beta,\delta,\varepsilon)$ denote the worst case running time of Algorithm 9 for input set T with $|T| = n$. For the sake of simplicity, we assume that n is a power of 2. Note that we always have $\kappa \leq n$.

If $\kappa = 0$, Algorithm 9 has running time $c_1 \in O(1)$. If $n \geq \kappa \geq 1$, Algorithm 9 has running time at most $c_2 \cdot (n \cdot T_\rho + n) \in O(n \cdot T_\rho)$ to obtain P, $T(n/2,\kappa,\beta,\delta,\varepsilon)$ for the recursive call in the pruning phase, $T_1(n,\beta,\delta,\varepsilon)$ to obtain the candidates, $C(n,\beta,\delta,\varepsilon)\cdot T(n,\kappa-1,\beta,\delta,\varepsilon)$ for the recursive calls in the candidate phase, one for each candidate, and $c_3 \cdot n \cdot T_\rho \cdot C(n,\beta,\delta,\varepsilon) \in O(n \cdot T_\rho \cdot C(n,\beta,\delta,\varepsilon))$

to eventually evaluate the candidate sets. Let $c = c_1 + c_2 + c_3 + 1$. We obtain the following recurrence relation:

$$T(n, \kappa, \beta, \delta, \varepsilon) \leq \begin{cases} c & \text{if } \kappa = 0 \\ C(n, \beta, \delta, \varepsilon) \cdot T(n, \kappa - 1, \beta, \delta, \varepsilon) + T(n/2, \kappa, \beta, \delta, \varepsilon) & \\ +T_1(n, \beta, \delta, \varepsilon) + cn \cdot T_\rho \cdot C(n, \beta, \delta, \varepsilon)) & \text{else} \end{cases}.$$

Let $f(n, \beta, \delta, \varepsilon) = \frac{1}{cn} \cdot T_1(n, \beta, \delta, \varepsilon) + T_\rho \cdot C(n, \beta, \delta, \varepsilon)$.

We prove that $T(n, \kappa, \beta, \delta, \varepsilon) \leq c \cdot 4^\kappa \cdot C(n, \beta, \delta, \varepsilon)^{\kappa+1} \cdot n \cdot f(n, \beta, \delta, \varepsilon)$, by induction on n, κ.

For $\kappa = 0$ we have $T(n, \kappa, \beta, \delta, \varepsilon) \leq c \leq cn \leq c \cdot 4^0 \cdot C(n, \beta, \delta, \varepsilon) \cdot n \cdot f(n, \beta, \delta, \varepsilon)$.

Now, let $n \geq \kappa \geq 1$ and assume the claim holds for $T(n', \kappa', \beta, \delta, \varepsilon)$, for each $\kappa' \in \{0, \ldots, \kappa - 1\}$ and $n' \in [n - 1]$. We have:

$$\begin{aligned} T(n, \kappa, \beta, \delta, \varepsilon) &\leq C(n, \beta, \delta, \varepsilon) \cdot T(n, \kappa - 1, \beta, \delta, \varepsilon) + T(n/2, \kappa, \beta, \delta, \varepsilon) \\ &\quad + T_1(n, \beta, \delta, \varepsilon) + cn \cdot T_\rho \cdot C(n, \beta, \delta, \varepsilon) \\ &\leq C(n, \beta, \delta, \varepsilon) \cdot c \cdot 4^{\kappa-1} \cdot C(n, \beta, \delta, \varepsilon)^\kappa \cdot n \cdot f(n, \beta, \delta, \varepsilon) \\ &\quad + c \cdot 4^\kappa \cdot C(n/2, \beta, \delta, \varepsilon)^{\kappa+1} \cdot \frac{n}{2} \cdot f(n/2, \beta, \delta, \varepsilon) \\ &\quad + cn \cdot f(n, \beta, \delta, \varepsilon) \\ &\leq \left(\frac{1}{4} + \frac{1}{2} + \frac{1}{4^\kappa C(n, \beta, \delta, \varepsilon)^{\kappa+1}}\right) c \cdot 4^\kappa \cdot C(n, \beta, \delta, \varepsilon)^{\kappa+1} \cdot n \cdot f(n, \beta, \delta, \varepsilon) \\ &\leq c \cdot 4^\kappa \cdot C(n, \beta, \delta, \varepsilon)^{\kappa+1} \cdot n \cdot f(n, \beta, \delta, \varepsilon). \end{aligned}$$

The last inequality holds, because $\frac{1}{4^\kappa C(n,\beta,\delta,\varepsilon)^{\kappa+1}} \leq \frac{1}{4}$, and the claim follows by induction. □

In the following, we apply Algorithm 9 to the problems of clustering polygonal curves under the Fréchet distance and point sequences under the dynamic time warping distance. Since Algorithm 9 approximates the generalized k-median problem, in both cases we obtain certain expressions of k-median clustering approximation algorithms for the respective input objects and distance measures.

4.2.3 Application to Polygonal Curves under the Fréchet Distance

Here, we apply Algorithm 9 to the problem of clustering polygonal curves under the Fréchet distance. We formally define the problem that we aim to approximate.

Problem Definition

Building upon the results in Section 3.1 we want to compute a set of k clusters that are best possibly described by their ℓ-median. Consequently, the ℓ-median problem (Problem 3.1.1) can be seen as a special case of the following problem for $k = 1$.

Problem 4.2.3 *The (k, ℓ)-**median clustering problem** is defined as follows, where $k \in \mathbb{N}$ and $\ell \in \mathbb{N}_{>1}$ are fixed (constant) parameters of the problem: given a set $T = \{\tau_1, \ldots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, compute a set $C \subset \mathbb{R}^d_\ell$ of k polygonal curves, such that* $\operatorname{cost}(T, C) = \sum_{i=1}^n \min_{c \in C} \mathrm{d_F}(\tau_i, c)$ *is minimal.*

Indeed, by setting $X = \mathbb{R}^d_*$, $Y = \mathbb{R}^d_m \subseteq X$, $Z = \mathbb{R}^d_\ell \subseteq X$ and $\rho = \mathrm{d}_\mathrm{F}$, Problem 4.2.3 is a specialization of the generalized k-median problem (Problem 4.1.1) and can therefore be approximated using Algorithm 9. We now review the related work.

Related Work

Only recently, Driemel et al. [87] introduced the (k, ℓ)-median and (k, ℓ)-center problem, where the latter is a similar extension of the ℓ-center problem to the clustering setting. Driemel et al. proved that (k, ℓ)-center as well as (k, ℓ)-median clustering is NP-hard when k is a part of the input and ℓ is fixed. Also, they showed that the doubling dimension of the metric space of polygonal curves under the Fréchet distance is unbounded, even when the complexity of the curves is bounded. For this reason, the k-median result by Ackermann et al. [6] for certain metric spaces of bounded doubling dimension can not be applied. However, the authors developed the first approximation schemes for these problems, for curves in $\mathbb{R}$. For the (k, ℓ)-median problem they circumvented the problem of the unbounded doubling dimension and proved that the sampling property indeed holds, so the k-median algorithm by Ackermann et al. can nevertheless be applied (for the (k, ℓ)-center problem they used a completely different algorithm). Following this work, Buchin et al. [59] developed the first constant factor approximation algorithm for (k, ℓ)-center clustering in $\mathbb{R}^d$. Furthermore, they provide improved results on the hardness of approximating (k, ℓ)-center clustering under the Fréchet distance: the (k, ℓ)-center problem is NP-hard to approximate within a factor of $(1.5 - \varepsilon)$ for curves in $\mathbb{R}$ and within a factor of $(2.25 - \varepsilon)$ for curves in $\mathbb{R}^d$, where $d \geq 2$, in both cases even if $k = 1$. Furthermore, for the (k, ℓ)-median variant, Buchin et al. [61] proved NP-hardness using a similar reduction. Again, the hardness holds even if k is equal to 1.

The (k, ℓ)-center and the (k, ℓ)-median problems have also been studied under other distance measures, like the discrete Fréchet and Hausdorff distance (a distance measure on point sets). Buchin et al. [59] provide the first constant factor approximation algorithm for (k, ℓ)-center clustering in $\mathbb{R}^d$ under the discrete Fréchet distance. Furthermore, they proved that the (k, ℓ)-center problem under the discrete Fréchet distance is NP-hard to approximate within a factor of $(2 - \varepsilon)$ for curves in $\mathbb{R}$ and within a factor of $(3 \sin \pi/3 - \varepsilon)$ for curves in $\mathbb{R}^d$, where $d \geq 2$. The hardness holds in both cases even if $k = 1$. Buchin et al. [61] developed the first $(1 + \varepsilon)$-approximation algorithms for (k, ℓ)-center and (k, ℓ)-median clustering in $\mathbb{R}^d$ under the discrete Fréchet distance. Furthermore, they presented an exact algorithm for (k, ℓ)-center clustering in $\mathbb{R}^2$ under the discrete Fréchet distance. Nath and Taylor [207] give the first near-linear time $(1 + \varepsilon)$-approximation algorithm for (k, ℓ)-median clustering under the discrete Fréchet distance and a novel $(1 + \varepsilon)$-approximation algorithm for (k, ℓ)-median clustering under the Hausdorff distance. They achieve these results by using the algorithm by Ackermann et al. [6] in conjunction with a generalization of doubling dimension that is satisfied by the discrete Fréchet and Hausdorff distance and implies the sampling property, which they name g-coverability. However, it is not known whether g-coverability holds for (k, ℓ)-median clustering under the continuous Fréchet and DTW distances.

We summarize the main algorithmic results on the (k, ℓ)-center and (k, ℓ)-median problem under the Fréchet distance.

Problem	Approx.	Running Time	Ambient	Measure	Reference
(k, ℓ)-center	$1 + \varepsilon$	$O(nm \log m)$, ε const.	$\mathbb{R}$	$\mathrm{d_F}$	[87]
	3	$O(nm \log m + m^3 \log m)$	$\mathbb{R}^2$		[59]
	6	$O(nm \log m + m^3 \log m)$	$\mathbb{R}^d$		
	1	$O((mn)^{2k\ell} m \log(mn))$	$\mathbb{R}^2$	$\mathrm{d_{dF}}$	[61]
	$1 + \varepsilon$	$O((\varepsilon^{-dk\ell} + \log n)mn)$	$\mathbb{R}^d$		
	3	$O(mn \log m)$			[59]
(k, ℓ)-median	$1 + \varepsilon$	$O(nm \log m)$, ε const.	$\mathbb{R}$	$\mathrm{d_F}$	[87]
	$1 + \varepsilon$ (bi-crit.)	$n \cdot 2^{O(\varepsilon^{-3} + \log m)}$	$\mathbb{R}^d$		Coro. 4.2.5
	$3 + \varepsilon$ (bi-crit.)				Coro. 4.2.4
	65	$O((n + \log^5 n)m \log m)$	$\mathbb{R}$		[87]
	109	$O(nm \log(m) + nm^3 \log m)$	$\mathbb{R}^d$		Thm. 4.3.8
	$1 + \varepsilon$	$nm \log^2(m) 2^{O(\varepsilon^{-1} \log \varepsilon^{-1})}$		$\mathrm{d_{dF}}$	[207]
		$O((m/\varepsilon)^{dk\ell} mn + mn \log^2 n)$			[61]

Approximation Algorithms

As we already hinted in Section 3.1.4, Algorithm 9 can approximate the (k, ℓ)-median problem for polygonal curves under the Fréchet distance, when provided with Algorithm 4 or Algorithm 5 as plugin algorithm. Note that it then computes a bi-criteria approximation, that is, the solution is approximated in terms of the cost *and* the number of vertices of the center curves, i.e., the centers come from $\mathbb{R}^d_{2\ell-2}$.

Our main results, which we state below, follow from Theorems 3.1.20 and 3.1.21, respectively Theorems 3.1.24 and 3.1.25, and Theorems 4.2.1 and 4.2.2. The first result is a $(3 + \varepsilon)$-approximation that follows by using Algorithm 4 as plugin and the second one is a $(1 + \varepsilon)$-approximation that follows by using Algorithm 5 as plugin.

Corollary 4.2.4 *Given two parameters $\delta, \varepsilon \in (0, 1)$ and a finite set $T \subset \mathbb{R}^d_m$ of polygonal curves, Algorithm 9 endowed with Algorithm 4 as MEDIAN-CANDIDATES and run with parameters $(T, \emptyset, k, \frac{20k^2}{\varepsilon} + 2k, \delta, \varepsilon/5)$ returns with probability at least $1 - \delta$ a set $C \subset \mathbb{R}^d_{2\ell-2}$ that is a $(3 + \varepsilon)$-approximate solution to the (k, ℓ)-median for T. Algorithm 9 then has running time $n \cdot 2^{O\left(\frac{\ln^2(1/\delta)}{\varepsilon^3} + \log(m)\right)}$.*

We note that the following result does not differ in asymptotic running time from the previous one. However, the hidden constants are larger if we combine Algorithm 9 and Algorithm 5.

Corollary 4.2.5 *Given two parameters $\delta \in (0, 1), \varepsilon \in (0, 0.158]$ and a finite set $T \subset \mathbb{R}^d_m$ of polygonal curves, Algorithm 9 endowed with Algorithm 5 as MEDIAN-CANDIDATES and run with parameters $(T, \emptyset, k, \frac{12k^2}{\varepsilon} + 2k, \delta, \varepsilon/3)$ returns with probability at least $1 - \delta$ a set $C \subset \mathbb{R}^d_{2\ell-2}$ that is a $(1 + \varepsilon)$-approximate solution to the (k, ℓ)-median for T. Algorithm 9 then has running time $n \cdot 2^{O\left(\frac{\ln^2(1/\delta)}{\varepsilon^3} + \log(m)\right)}$.*

Finally, we note that from a practical point of view, both results yield inefficient algorithms due to the exponential dependency in ε and k and the high polynomial dependency in m.

4.2.4 Application to Point Sequences under the Dynamic Time Warping Distance

Here, we apply Algorithm 9 to the problem of clustering point sequences under the dynamic time warping distance. We formally define the problem that we aim to approximate.

Problem Definition

Building upon the results in Section 3.2 we want to compute a set of k clusters that are best possibly described by their restricted (p, q)-mean. Consequently, the restricted (p, q)-mean problem (Problem 3.2.2) can be seen as a special case of the following problem for $k = 1$.

Problem 4.2.6 *The (k, ℓ, p, q)-**mean clustering problem** is defined as follows, where $k \in \mathbb{N}$, $\ell \in \mathbb{N}_{>1}$ and $p, q \in [1, \infty)$ are fixed (constant) parameters of the problem: given a set $T = \{\tau_1, \dots, \tau_n\} \subseteq M^{\leq m}$ of point sequences, compute a set $C \subseteq M^{\leq \ell}$ of k point sequences, such that $\operatorname{cost}_p^q(T, C) = \sum_{i=1}^n \min_{c \in C} \mathrm{d}_{\mathrm{DTW}_p}(c, \tau_i)^q$ is minimal.*

Approximating a (k, ℓ, p, q)-mean clustering problem is equivalent to approximating the generalized k-median clustering problem, where the distance between any center c and any input point sequence τ is measured by the non-metric distance function $\rho(\tau, c) = \mathrm{d}_{\mathrm{DTW}_p}(\tau, c)^q$. Therefore, Problem 4.2.6 is compatible with Problem 4.1.1 by setting $X = M^*$, $Y = M^{\leq m} \subseteq X$ and $Z = M^{\leq \ell} \subseteq X$. Consequently, we can approximate it using Algorithm 9. We now review the related work.

Related Work

To the best of our knowledge, clustering under the dynamic time warping distance has so far only been considered in practice, not in theory. In particular, there do not exist clustering algorithms for point sequences under (p-)DTW that admit formal guarantees on the quality of the returned solution. Recall that even for the base case ($k = 1$), almost no algorithms with such a formal guarantee exist, cf. Section 3.2.

In practice, however, clustering under DTW is popular, but heuristics are prevalent. Often, generic local search algorithms are used that select their centers from the input (e.g. k-medoids, which is similar to k-median, but the centers are restricted to come from the input), but a k-means style averaging approach using DBA (DTW barycenter averaging, see Section 3.2.2) has also been considered early [215]. We note that the latter allows for centers of arbitrary but predefined complexity that do not need to come from the input, which is in line with the $(k, \ell, 2, 2)$-mean problem (Problem 4.2.6) and this approach has also been extended to fuzzy k-medoids and k-means [152], which are extensions of k-medoids and k-means where each element is not strictly assigned to a cluster.

Only recently, clustering under the continuous dynamic time warping distance has been considered [44]. This distance measure is a continuous extension of the dynamic time warping distance – similar to Fréchet distance, an implicit linear interpolation between consecutive points of a sequence is introduced –, which aims at a better robustness towards time series measured with vastly different sampling rates. The clustering approach is similar to the approach taken in [215] and an adaption of DBA, called CDBA, is also provided.

Approximation Algorithms

First, we extend the methods developed in Section 3.2.4. The following algorithm is an adaptation of Algorithm 6 that is able to return candidates, containing with high probability an approximate restricted p-mean for a subset $T' \subseteq T$ of certain size, as required by Algorithm 9. Recall, that the approximation factor is $(2^p + \varepsilon)$.

Algorithm 10 $(1, \ell, p, p)$-Mean Clustering Approximate Candidates

```
procedure CAND($T = \{\tau_1 = (\tau_{1,1}, \dots, \tau_{1,|\tau_1|}), \dots, \tau_n = (\tau_{n,1}, \dots, \tau_{n,|\tau_n|})\}, \beta, \delta, \varepsilon, p$)
    $P \leftarrow \bigcup_{i=1}^{n} \bigcup_{j=1}^{|\tau_i|} \{\tau_{i,j}\}$
    $S \leftarrow$ sample $\lceil (2^p/\varepsilon + 1)\beta m \ln(\ell/\delta) \rceil$ points from $P$ uniformly and independently at
        random with replacement
    return $S^{\leq \ell}$
```

We prove the correctness and analyze the running time of Algorithm 10.

Theorem 4.2.7 *Given a finite set $T \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ) and parameters $\beta \in [1, \infty)$, $\delta \in (0,1)$, $\varepsilon \in (0, \infty)$ and $p \in [1, \infty)$, with probability at least $1 - \delta$ the set of candidates that Algorithm 10 returns contains a $(2^p + \varepsilon)$-approximate restricted p-mean for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$. Algorithm 10 returns $O((\beta\varepsilon^{-1} m \ln(1/\delta))^\ell)$ candidates and if the time needed to evaluate ϑ is constant, then it has running time $O((\beta\varepsilon^{-1} m \ln(1/\delta))^\ell)$.*

Proof. By Lemma 3.2.6 applied on T', we have that there exist balls $B_1, \dots, B_{\ell'} \subseteq P$, $\ell' \leq \ell$, each of cardinality at least $\left(\frac{\varepsilon}{2^p + \varepsilon}\right) \cdot |T'|$ such that any point sequence $c = (c_1, \dots, c_{\ell'})$ with $c_i \in B_i$ for all $i \in [\ell']$, is a $(2^p + \varepsilon)$-approximate restricted p-mean of T'. We upper bound the probability that S does not contain any point from a fixed B_i:

$$\Pr\left[|B_i \cap S| = 0\right] \leq \left(\frac{|P| - \left(\frac{\varepsilon}{2^p + \varepsilon}\right) \cdot |T'|}{|P|}\right)^{|S|} \leq \left(1 - \left(\frac{\varepsilon}{(2^p + \varepsilon)\beta m}\right)\right)^{|S|} \leq \frac{\delta}{\ell}.$$

By a union bound we have $\Pr[\min_{i \in [\ell']} |B_i \cap S| < 1] \leq \delta$. Hence, with probability at least $1 - \delta$, there is a point sequence $c \in S^{\leq \ell}$ which is a $(2^p + \varepsilon)$-approximate restricted p-mean of T'.

The number of candidates that Algorithm 10 returns, as well as its running time, is $|S^{\leq \ell}| \in O((\beta\varepsilon^{-1} m \ln(1/\delta))^\ell)$. □

The following corollary follows from Theorem 4.2.7 and Theorems 4.2.1 and 4.2.2. We note that the achieved approximation factor in this case degrades to $(2^{p+1} + \varepsilon)$, due to the use of Algorithm 9.

Corollary 4.2.8 *Given three parameters $\delta \in (0,1), \varepsilon \in (0, \infty), p \in [1, \infty)$ and a finite set $T \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ), Algorithm 9 endowed with Algorithm 10 (with parameter p) as* MEDIAN-CANDIDATES *and run with parameters $(T, \emptyset, k, \frac{8k^2}{\varepsilon} + 2k, \delta, \varepsilon/2)$ returns with probability at least $1 - \delta$ a set $C \subseteq M^{\leq \ell}$ that is a $(2^{p+1} + \varepsilon)$-approximate solution to the (k, ℓ, p, p)-mean for T. If the time needed to evaluate ϑ is constant, then Algorithm 9 has running time $O(n(m\varepsilon^{-2} \ln(1/\delta))^{2\ell(k+2)})$.*

Again, in the Euclidean case we can derandomize the algorithm and obtain together with Algorithm 9 a deterministic approximation algorithm for the (k, ℓ, p, p)-mean problem. The following algorithm is an adaption of Algorithm 7. Recall that $\mathcal{B}$ is the set of all metric balls with respect to (M, ϑ), which is the Euclidean space in this case.

Algorithm 11 $(1, \ell, p, p)$-Mean Clustering Approximate Candidates (Deterministic)

```
procedure CAND-D($T = \{\tau_1 = (\tau_{1,1}, \ldots, \tau_{1,|\tau_1|}), \ldots, \tau_n = (\tau_{n,1}, \ldots, \tau_{n,|\tau_n|})\}, \varepsilon, p$)
    $\varepsilon' \leftarrow \frac{\varepsilon}{2^{p-1}+\varepsilon}$, $P \leftarrow \bigcup_{i=1}^{n} \bigcup_{j=1}^{|\tau_i|} \{\tau_{i,j}\}$
    $S \leftarrow$ compute an $(\varepsilon'/(m\beta))$-net of $(P, \mathcal{B}_{|P})$
    return $S^{\leq \ell}$
```

The correctness and running time of Algorithm 11 follow from Definition 2.6.5, Lemmas 3.2.6 and 3.2.9, and Theorem 2.6.7.

Theorem 4.2.9 *Given a finite set $T \subset \left(\mathbb{R}^d\right)^{\leq m}$ of n point sequences and parameters $\beta \in [1, \infty)$ $\varepsilon \in (0, \infty)$, $p \in [1, \infty)$, the set of candidates that Algorithm 11 returns contains a $(2^p + \varepsilon)$-approximate restricted p-mean for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$. Algorithm 10 returns $O\left(\left(\frac{\beta m}{\varepsilon} \log \frac{\beta m}{\varepsilon}\right)^{\ell}\right)$ candidates and has running time*[1] $O\left(nm\left(\left(\frac{\beta m}{\varepsilon} \log \frac{\beta m}{\varepsilon}\right)^{d+1} + \left(\frac{\beta m}{\varepsilon} \log \frac{\beta m}{\varepsilon}\right)^{\ell}\right)\right)$.

Proof. By Lemma 3.2.6 applied on T', we have that there exist balls $B_1, \ldots, B_{\ell'} \subseteq P$, $\ell' \leq \ell$, each of cardinality at least $\left(\frac{\varepsilon}{2^p+\varepsilon}\right) \cdot |T'|$ such that any point sequence $c = (c_1, \ldots, c_{\ell'})$ with $c_i \in B_i$ for all $i \in [\ell']$, is a $(2^p + \varepsilon)$-approximate restricted p-mean of T'. Since we compute an $(\varepsilon'/(m\beta))$-net of P and $|P| \leq nm$, S contains at least one point from each of $B_1, \ldots, B_{\ell'}$ by Definition 2.6.5. Hence, $S^{\leq \ell}$ contains a $(2^p + \varepsilon)$-approximate restricted p-mean for T.

The VC dimension of the range space $(P, \mathcal{B}_{|P})$ is bounded by $d+1$, see [130]. By Lemma 3.2.9, we can use Theorem 2.6.7 to compute an $(\varepsilon'/(\beta m))$-net S of $(P, \mathcal{B}_{|P})$, with size $|S| \in O\left(\frac{m\beta}{\varepsilon} \log\left(\frac{m\beta}{\varepsilon}\right)\right)$, in time $O\left(nm\left(\frac{\beta m}{\varepsilon} \log\left(\frac{\beta m}{\varepsilon}\right)\right)^{d+1}\right)$. □

The following corollary follows from Theorem 4.2.9 and Theorems 4.2.1 and 4.2.2.

Corollary 4.2.10 *Given two parameters $\varepsilon \in (0, \infty), p \in [1, \infty)$ and a finite set $T \subset \left(\mathbb{R}^d\right)^{\leq m}$ of point sequences, Algorithm 9 endowed with Algorithm 11 (with parameter p) as* MEDIAN-CANDIDATES *and run with parameters $(T, \emptyset, k, \frac{8k^2}{\varepsilon} + 2k, \delta, \varepsilon/2)$ returns a set $C \subset \left(\mathbb{R}^d\right)^{\leq \ell}$ that is a $(2^{p+1} + \varepsilon)$-approximate solution to the (k, ℓ, p, p)-mean for T. Algorithm 9 has running time $O\left(nm\left(\frac{m}{\varepsilon^2} \log\left(\frac{m}{\varepsilon^2}\right)\right)^{(k+1)\ell+d+2}\right)$.*

Now, we use the methods developed in Section 3.2.5. The following algorithm is inspired by Algorithm 8 and is able to return candidates, containing an approximate restricted $(p, 1)$-mean for a subset $T' \subseteq T$ of certain size, as required by Algorithm 9. Recall that the corresponding problem asks for a median point sequence under p-DTW of complexity at most ℓ. We achieve an approximation factor of $8(m\ell)^{1/p}$. We note that Algorithm 8 could not be extended, unfortunately, since we would need an upper and a lower bound on the cost of the optimal median point sequence of the subset $T' \subseteq T$ to set up the grids that the algorithm uses. The techniques we use in

[1] We assume d to be constant.

Section 3.1.4 to obtain such a lower bound, however, rely on the triangle inequality, which does not hold under p-DTW.

The main idea of the algorithm is that one can use random sampling and approximate simplifications to obtain a simple algorithm for computing a set of candidates.

Algorithm 12 $(1, \ell, p, 1)$-Clustering Approximate Candidates

```
procedure CAND($T = \{\tau_1, \dots, \tau_n\}, \beta, \delta, p$)
    $S \leftarrow$ sample $\lceil 2\beta \cdot \ln(2/\delta) \rceil$ point sequences from $T$ uniformly and independently at
        random with replacement
    $C \leftarrow \emptyset$
    for each $\tau \in S$ do
        $\tau' \leftarrow$ 2-approximate minimum-error $\ell$-simplification of $\tau$, under $d_{\mathrm{DTW}_p}$ (Algorithm 2)
        $C \leftarrow C \cup \{\tau'\}$
    return $C$
```

Theorem 4.2.11 *Given a finite set $T \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ) and parameters $\beta \in [1, \infty)$, $\delta \in (0, 1)$ and $p \in [1, \infty)$, with probability at least $1 - \delta$ the set of candidates that Algorithm 12 returns contains a $(8 \cdot (m\ell)^{\frac{1}{p}})$-approximate restricted $(p, 1)$-mean for any $T' \subseteq T$, if $|T'| \geq \frac{1}{\beta}|T|$. Algorithm 12 returns $O(\beta \ln(1/\delta))$ candidates and if the time needed to evaluate ϑ is constant, then it has running time $O(\beta m^4 \ell \ln(1/\delta))$.*

Proof. We assume that $|T'| \geq \frac{1}{\beta}|T|$. We use a Chernoff bound (cf. Theorem 2.4.17) to upper bound the probability that $|S \cap T'| \leq |S|/(2\beta)$. Notice that $|S \cap T'|$ is the sum of independent Bernoulli trials and $\mathrm{Exp}[|S \cap T'|] \geq |S| \cdot \beta^{-1}$. Hence,

$$\Pr\left[|S \cap T'| \leq \frac{|S|}{2\beta}\right] \leq \exp\left(-\frac{|S|}{8\beta}\right) \leq \frac{\delta}{2}.$$

In other words, with probability at most $\delta/2$ no subset $S' \subseteq S$, of cardinality at least $\frac{|S|}{2\beta}$, is a subset of T'. We condition the rest of the proof on the contrary event, denoted by $\mathcal{E}_{T'}$, namely, that there is a subset $S' \subseteq S$ with $S' \subseteq T'$ and $|S'| \geq \frac{|S|}{2\beta}$. Note that S' is then a uniform and independent sample of T' (see Section 2.4.1).

In the following, let τ' denote a $(2, \ell)$-simplification of any point sequence $\tau \in S'$. Then, by Theorem 3.2.11

$$\mathrm{Exp}\Big[\mathrm{cost}_p^1(T', \tau') \mid \mathcal{E}_{T'}\Big] \leq 4m^{1/p}\ell^{1/p} \cdot \mathrm{cost}_p^1(T', c^*),$$

where c^* is an optimal restricted $(p, 1)$-mean for T'. By Markov's inequality,

$$\Pr\left[\mathrm{cost}_p^1(T', \tau') \geq 8m^{1/p}\ell^{1/p} \cdot \mathrm{cost}_p^1(T', c^*) \mid \mathcal{E}_{T'}\right] \leq \frac{1}{2}.$$

Hence, by independence of the random sampling,

$$\Pr\left[\min_{\tau \in S \cap T'} \mathrm{cost}_p^1(T', \tau') \geq 8m^{1/p}\ell^{1/p} \cdot \mathrm{cost}_p^1(T', c^*) \mid \mathcal{E}_{T'}\right] \leq \frac{1}{2^{|S|/(2\beta)}} \leq \frac{\delta}{2}.$$

The claim on the approximation quality now follows by an application of Proposition 2.4.5.

Using Algorithm 2 to compute simplifications, Algorithm 12 needs $O(\beta m^4 \ell \ln(1/\delta))$ time to compute C (see Theorem 2.8.4). The number of returned candidates is $O(\beta \ln(1/\delta))$. □

The following corollary follows from Theorem 4.2.11 and Theorems 4.2.1 and 4.2.2.

Corollary 4.2.12 *Given two parameters $\delta \in (0,1), p \in [1,\infty)$ and a finite set $T \subseteq M^{\leq m}$ of point sequences over some metric space (M, ϑ), Algorithm 9 endowed with Algorithm 12 (with parameter p) as* Median-Candidates *and run with parameters $(T, \emptyset, k, \frac{48k^2}{\varepsilon} + 2k, \delta, 0)$ returns with probability at least $1-\delta$ a set $C \subseteq M^{\leq \ell}$ that is a $(8+\varepsilon)(m\ell)^{\frac{1}{p}}$-approximate solution to the $(k,\ell,p,1)$-mean for T. If the time needed to evaluate ϑ is constant, then Algorithm 9 has running time $O(nm^4(\varepsilon^{-2}\ln^2(1/\delta))^{k+2})$.*

We have presented approximation algorithms for the problems of (k,ℓ)-median clustering of polygonal curves under the Fréchet distance and (k,ℓ,p,p)-mean clustering, as well as $(k,\ell,p,1)$-mean clustering, of point sequences under the dynamic time warping distance. While these algorithms are highly developed – in the sense that all available insights on the problem have been incorporated to optimize upon achieved approximation factors and running times – all suffer from high and unpractical running times. Optimizing these algorithms towards a better running time, or even designing new algorithms with better running times, requires gaining more insights (or combining the available insights differently, if possible) on the problem. The experience is that this often requires a lot of effort, while yielding only small improvements.

Another more vital approach is to pre-process the given data, such that existing algorithms behave similarly on the pre-processed data, while taking less running time. This may be achieved by cleverly reducing the amount of data, while keeping the important parts of the data. This and other approaches are studied in the field of ε-coresets, which we now present. First, we note that the following results strongly depend on metric properties and are therefore unfortunately not applicable to clustering under p-DTW.

4.3 Coresets for Clustering in Metric Spaces

The observation underlying coresets is that for many problems a solution to the problem is completely determined by a subset of the given input set or a set of a certain size that may be much smaller than the given input set. For example, the minimum enclosing ball, respectively minimum enclosing sphere, of a point set in $\mathbb{R}^d$ is completely determined by a subset of at most $d+1$ points [96]. A further example is the problem of computing the directional width of a finite point set $P \subset \mathbb{R}^d$. The directional width is the extent of P when projected onto a line through the origin. Formally, any such line is determined by a unit vector $v \in \mathbb{R}^d$ and the directional width is simply $\max_{p \in P}\langle p, v\rangle - \min_{q \in P}\langle q, v\rangle$. Given such a unit vector v, it takes time $O(|P|)$ to naïvely compute the directional width. However, if we want to compute the directional width for several directions, this naïve algorithm can be improved to run in time $O(\varepsilon^{-d+1})$ if we accept a relative error in $[(1-\varepsilon), 1]$ and one pre-processing with running time $O(|P| + \varepsilon^{-d})$. In this case we can compute a suitable bounding box B of P, which is possible in time $O(|P|)$, and cover B with a uniform grid G of $O(\varepsilon^{-d})$ cells. Finally, we run the naïve algorithm on a set S of $O(\varepsilon^{-d+1})$ points of P determined by the cells of G, instead of P itself, and obtain the desired approximation [9, 130]. This approach is particularly successful if $\varepsilon^{-d+1} \ll |P|$.

Indeed, the aforementioned set S is an example of an ε-coreset. In general, coresets are a form of problem specific pre-processing: they capture the structure of the input set that is relevant for the given problem, are substantially smaller than the input set and are meant to serve as a proxy to run an existing algorithm on. As described above, this may yield a considerable speedup, even if a brute force algorithm is utilized. Often, coresets carry additional structure, like weights or constants, which are then built into the objective function one wants to approximate. This largely depends on the application at hand and there are indeed applications for which sub-linear size coresets generally do not exist, see e.g. [128, 206]. Since their recent formal introduction by Har-Peled and Mazumdar [133], coresets have become very popular and indeed, several notions of coresets have established. On one side, we have weak and strong coresets. This notion concerns the behavior of the coreset towards a solution to the problem. Strong coresets guarantee a certain approximation for all possible solutions, while weak coresets yield this guarantee only for certain solutions, cf. [205]. Another notion concerns the approximation error itself, which can either be relative, in this case we have a multiplicative coreset, or absolute, in this case we have an additive coreset. The canonical notion of an ε-coreset is arguably the multiplicative strong coreset, which is also the notion coresets have been introduced with.

The techniques used to obtain coresets can mainly be partitioned into two categories: input filtering and geometric constructions. As the name suggests, in the first category one computes coresets by picking (important) elements from the given input while ignoring the remainder. Here, sampling techniques are prevalent. For certain problems, uniform sampling can luckily be applied. However, for most problems this is not the case. Here, a custom probability distribution needs to be computed. Since this approach is versatile and particularly successful, a whole framework, named *sensitivity sampling* or *importance sampling*, has formed around it [175, 101]. The other category comprises techniques that leverage properties of the input space, i.e., the space underlying the given input. Here, coverings, like ε-ball covers or notions of grids, like uniform or exponential grids, are popular. Probably most constructions utilize sampling, though.

Before its formal introduction in 2004, the concept of ε-coresets has been applied to some geometric problems [9, 8, 127], including clustering problems. Indeed, the breakthrough of ε-coresets was the application to Euclidean k-median and k-means clustering by Har-Peled and Mazumdar [133], which implied the first randomized linear time $(1+\varepsilon)$-approximation algorithms for these problems (with ε and k fixed). Furthermore, Har-Peled and Mazumdar showed that their ε-coresets can be used to maintain $(1+\varepsilon)$-approximate k-median, respectively k-means, solutions in the insertion-only streaming setting, using polylogarithmic space and update time.

Following the work of Har-Peled and Mazumdar, the coreset literature started to flourish. Since then, coresets have widely been applied, e.g. to clustering problems [131, 129, 102, 112, 103, 5, 72, 106, 110, 238, 146, 30], statistical problems [79, 248, 105, 26, 188, 259, 148, 260, 217, 206, 48, 147, 73, 163, 258, 242], geometrical problems [128, 111, 92, 10, 7, 143, 119, 104, 2], machine learning problems [137, 249, 107, 99, 186, 224, 166, 224, 186, 241, 40, 187, 201], optimization [74, 222, 240, 145] and many more.

In this work, we focus on coresets for *generalized k-median clustering*, with the restriction that the underlying space is an *arbitrary* metric space. In particular, this captures the (k,ℓ)-median problem for polygonal curves. We start with a formal definition of these ε-coresets, then we review the related work.

4.3.1 Problem Definition

We formally define ε-coresets for generalized k-median clustering (recall Problem 4.1.1). However, in this section we require that the underlying space $\mathcal{X} = (X, \rho)$ is a *metric* space.

Problem 4.3.1 *Given $\varepsilon \in (0,1)$ and a finite non-empty set $T \subseteq Y$, compute a (multi-)set $S \subseteq X$ together with a weight function $w \colon S \to \mathbb{R}_{>0}$, such that for all $C \subseteq Z$ with $|C| = k$ it holds that*

$$(1-\varepsilon)\operatorname{cost}(T,C) \leq \operatorname{cost}_w(S,C) \leq (1+\varepsilon)\operatorname{cost}(T,C),$$

where $\operatorname{cost}_w(S,C) = \sum_{s \in S} w(s) \cdot \min_{c \in C} \rho(s,c)$.

S together with w is called a weighted ε-coreset for generalized k-median clustering of T. In the following, we carefully review the related work.

4.3.2 Related Work

Langberg and Schulman [175] developed a framework for computing relative error approximations of integrals over any function from a given family of unbounded and non-negative real functions. In particular, this framework can be used to compute ε-coresets for k-median and k-means clustering of points in $\mathbb{R}^d$ with objective functions based on sums of distances (induced by a norm) among the points and their closest center. The idea of their framework is to sub-sample the input with respect to a certain non-uniform probability distribution, which is computed using an approximate solution to the problem. More precisely, the approximate solution is used to compute an upper bound on the sensitivity of each data element. The sensitivity is the maximum fraction of cost that the element may cause for any possible solution. It is a notion of the data elements *importance* for the problem and the probability distribution is set up such that each element has probability proportional to its importance. A sample of a certain size drawn from this distribution and properly weighted, is an ε-coreset for the underlying clustering problem with high probability.

Follwing this work, Feldman and Langberg [101] developed a unified framework for approximate clustering, which is largely based on ε-coresets. They combine the techniques by Langberg and Schulman with ε-approximations, which stem from the framework of range spaces and VC dimension developed in statistical learning theory. In result, they address a spectrum of clustering problems, such as k-median clustering of points and lines in $\mathbb{R}^d$, projective clustering of points in $\mathbb{R}^d$ and also other problems like Euclidean subspace approximation and k-median clustering in *finite* metric spaces. We note that using the latter is far from optimal for our problem of (k,ℓ)-median clustering. Using these coresets means that we would have to choose our centers from the input, or another predefined finite set, like a set of simplifications of the input. Since the input curves may be corrupted by noise or sampling artifacts, and simplifications can be sensitive to such artifacts, this may lead to bad centers in the subsequent clustering computation.

Braverman et al. [46] improved the aforementioned framework by switching to (ε,η)-approximations, which leads to substantially smaller sample sizes in many cases. Also, they simplified and further generalized the framework and applied it to k-means clustering of points in $\mathbb{R}^d$.

Feldman et al. [108] modified the approach by Braverman et al. by using another range space, thereby obtaining small coresets for k-means of points and lines in $\mathbb{R}^d$, and j-dimensional affine subspace k-clustering problems. Also, their coresets yield an improved streaming algorithm for Euclidean k-means clustering.

To the best of our knowledge, there are no results in the literature on ε-coresets for clustering problems on curves and in the following, we develop an approach that is capable of computing ε-coresets for the (k, ℓ)-median problem on polygonal curves. We achieve this by applying a variant of the sensitivity sampling framework. Our first step is to bound the sensitivity of each $\tau \in T$. That is the maximum fraction of $\operatorname{cost}(T, C)$ caused by τ, for all possible solutions C.

4.3.3 Sensitivity Bound

We analyze the problem in terms of functions. This allows us to apply the improved sensitivity sampling framework by Braverman et al. [46]. Therefore, given a set $T = \{\tau_1, \dots, \tau_n\} \subseteq Y$ we define $F = \{f_1, \dots, f_n\}$ to be a set of functions with $f_i \colon 2^Z \setminus \{\emptyset\} \to \mathbb{R}_{\geq 0},\ C \mapsto \min_{c \in C} \rho(c, \tau_i)$. For each $C \in 2^Z \setminus \{\emptyset\}$ we now have $\operatorname{cost}(T, C) = \sum_{i=1}^n f_i(C)$.

We give a universal bound on the sensitivities of the input elements. Intuitively, these are the maximum fraction of cost that an element may cause for any possible solution to the problem. We formally define the sensitivities of the inputs $\tau \in T$ in terms of the respective functions and to comply with the generalized k-median problem we only take into account the k-subsets $C \subseteq Z$.

Definition 4.3.2 [101] *Let F be a finite and non-empty set of functions $f \colon 2^Z \setminus \{\emptyset\} \to \mathbb{R}_{\geq 0}$. For $f \in F$ we define the sensitivity with respect to F:*

$$\mathfrak{s}(f, F) = \sup_{\substack{C = \{c_1, \dots, c_k\} \subseteq Z \\ \sum\limits_{g \in F} g(C) > 0}} \frac{f(C)}{\sum\limits_{g \in F} g(C)}.$$

We define the total sensitivity of F as $\mathfrak{S}(F) = \sum_{f \in F} \mathfrak{s}(f, F)$.

We now prove a bound on the sensitivity of all $f \in F$, which then yields a bound on the total sensitivity of F. To compute the bound, any (bi-criteria) approximate solution to the generalized k-median problem can be used. Our analysis is an adaption of the analysis of the sensitivities for sum-based k-clustering of points in $\mathbb{R}^d$, by Langberg and Schulman [175].

Lemma 4.3.3 *Let $k' \in \mathbb{N}$, $C^* = \{c_1^*, \dots, c_k^*\} \subseteq Z$ with $\Delta^* = \sum_{i=1}^n f_i(C^*)$ minimal and $\hat{C} = \{\hat{c}_1, \dots, \hat{c}_{k'}\} \subseteq X$ with $\hat{\Delta} = \sum_{i=1}^n f(\hat{C}) \leq \alpha \cdot \Delta^*$ for an $\alpha \in [1, \infty)$. Breaking ties arbitrarily, we assume that every $\tau \in T$ has a unique nearest neighbor in $\hat{C}$ and for $i \in [k']$, we define $\hat{V}_i = \{\tau \in T \mid \forall j \in [k'] : \rho(\tau, \hat{c}_i) \leq \rho(\tau, \hat{c}_j)\}$ to be the Voronoi cell of $\hat{c}_i$ and $\hat{\Delta}_i = \sum_{\tau \in \hat{V}_i} \rho(\tau, \hat{c}_i)$ to be its cost. For each $i \in [k']$ and $\tau_j \in \hat{V}_i$ it holds that*

$$\gamma(f_j) = \left(1 + \sqrt{\frac{2k'}{3\alpha}}\right) \left(\frac{\alpha \rho(\tau_j, \hat{c}_i)}{\hat{\Delta}} + \frac{2\alpha \hat{\Delta}_i}{\hat{\Delta} |\hat{V}_i|}\right) + \left(1 + \sqrt{\frac{3\alpha}{2k'}}\right) \frac{2}{|\hat{V}_i|} \geq \mathfrak{s}(f_j, F)$$

and $\Gamma = \sum_{f \in F} \gamma(f) = 2k' + 2\sqrt{6\alpha k'} + 3\alpha \geq \mathfrak{S}(F)$.

Proof. We assume that $\hat{\Delta} > 0$. By assumption $\hat{V}_1, \dots, \hat{V}_{k'}$ form a partition of T and by definition $\hat{\Delta} = \sum_{i=1}^{k'} \hat{\Delta}_i$ as well as $\sum_{f \in F} f(C) \geq \hat{\Delta}/\alpha$ for each $C = \{c_1, \dots, c_k\} \subseteq Z$. For $i \in [k']$, we let $\hat{B}_i = \{\tau \in \hat{V}_i \mid \rho(\tau, \hat{c}_i) \leq 2\hat{\Delta}_i/|\hat{V}_i|\}$. It holds that $|\hat{B}_i| \geq |\hat{V}_i|/2$, since otherwise

$\sum_{\tau \in \hat{V}_i \setminus \hat{B}_i} \rho(\tau, \hat{c}_i) > \hat{\Delta}_i$, which is a contradiction. By the triangle inequality, we have for each $\{c_1, \dots, c_k\} \subseteq Z$, $i \in [k']$, $j \in [k]$ and $\tau \in T$:

$$\rho(\hat{c}_i, c_j) \leq \rho(\hat{c}_i, \tau) + \rho(\tau, c_j) \iff \rho(\tau, c_j) \geq \rho(\hat{c}_i, c_j) - \rho(\hat{c}_i, \tau).$$

Furthermore, since ρ is non-negative: $\rho(\tau, c_j) \geq \max\{0, \rho(\hat{c}_i, c_j) - \rho(\hat{c}_i, \tau)\}$.

For each $C = \{c_1, \dots, c_k\} \subseteq Z$, $i \in [k']$ and $\beta \in [0, 1]$ we now have the following bound:

$$\sum_{f \in F} f(C) \geq \beta \sum_{\tau \in \hat{B}_i} \min_{j \in [k]} \rho(\tau, c_j) + (1 - \beta)\frac{\hat{\Delta}}{\alpha} \geq \beta \max\left\{0, \rho(\hat{c}_i, c_j) - \frac{2\hat{\Delta}_i}{|\hat{V}_i|}\right\} \frac{|\hat{V}_i|}{2} + (1 - \beta)\frac{\hat{\Delta}}{\alpha},$$

where among $\{c_1, \dots, c_k\}$, c_j is closest to $\hat{c}_i$. Using the triangle inequality and the above bound yields for each $\beta \in [0, 1)$, $i \in [k']$ and $\tau_m \in \hat{V}_i$:

$$\begin{aligned}
\mathfrak{s}(f_m, F) &\leq \sup_{\{c_1, \dots, c_k\} \subseteq Z} \frac{\rho(\tau_m, \hat{c}_i) + \rho(\hat{c}_i, c_j)}{\beta \max\left\{0, \rho(\hat{c}_i, c_j) - \frac{2\hat{\Delta}_i}{|\hat{V}_i|}\right\} \frac{|\hat{V}_i|}{2} + (1 - \beta)\frac{\hat{\Delta}}{\alpha}} \\
&\leq \sup_{\substack{\{c_1, \dots, c_k\} \subseteq Z \\ \rho(\hat{c}_i, c_j) \geq 2\hat{\Delta}_i / |\hat{V}_i|}} \frac{\rho(\tau_m, \hat{c}_i) + \rho(\hat{c}_i, c_j)}{\beta\left(\frac{|\hat{V}_i|\rho(\hat{c}_i, c_j)}{2} - \hat{\Delta}_i\right) + (1 - \beta)\frac{\hat{\Delta}}{\alpha}}
\end{aligned}$$

Here, again c_j is closest to $\hat{c}_i$ among $\{c_1, \dots, c_k\}$ and the last inequality follows because it can be observed that the term takes smaller values for $\rho(\hat{c}_i, c_j) < 2\hat{\Delta}_i/|\hat{V}_i|$ than for $\rho(\hat{c}_i, c_j) \geq 2\hat{\Delta}_i/|\hat{V}_i|$, independent of β. Now, to obtain a bound that is independent of c_j, we substitute $\rho(\hat{c}_i, c_j)$ by a free variable x and let

$$h\colon [2\hat{\Delta}_i/|\hat{V}_i|, \infty) \to \mathbb{R}_{\geq 0},\ x \mapsto \frac{\rho(\tau_m, \hat{c}_i) + x}{\beta\left(\frac{|\hat{V}_i|x}{2} - \hat{\Delta}_i\right) + (1 - \beta)\frac{\hat{\Delta}}{\alpha}}.$$

The derivative of h is

$$\frac{(1 - \beta)\frac{\hat{\Delta}}{\alpha} - \beta\left(\frac{|\hat{V}_i|\rho(\tau_m, \hat{c}_i)}{2} + \hat{\Delta}_i\right)}{\left(\beta\left(\frac{|\hat{V}_i|x}{2} - \hat{\Delta}_i\right) + (1 - \beta)\frac{\hat{\Delta}}{\alpha}\right)^2}$$

and it can be observed that the sign of this function is independent of x. Therefore, h is a monotonic function and is thus either maximized at $x = 2\hat{\Delta}_i/|\hat{V}_i|$ or when $x \to \infty$. Using l'Hôspital's rule we obtain

$$\mathfrak{s}(f_m, F) \leq \max\left\{\frac{\rho(\tau_m, \hat{c}_i) + \frac{2\hat{\Delta}_i}{|\hat{V}_i|}}{(1 - \beta)\frac{\hat{\Delta}}{\alpha}}, \frac{1}{\beta\frac{|\hat{V}_i|}{2}}\right\} \leq \frac{\alpha\rho(\tau_m, \hat{c}_i)}{(1 - \beta)\hat{\Delta}} + \frac{2\alpha\hat{\Delta}_i}{(1 - \beta)\hat{\Delta}|\hat{V}_i|} + \frac{2}{\beta|\hat{V}_i|}.$$

Therefore,

$$\mathfrak{S}(F) \leq \sum_{i=1}^{k'} \sum_{\tau_m \in \hat{V}_i} \left(\frac{\alpha\rho(\tau_m, \hat{c}_i)}{(1 - \beta)\hat{\Delta}} + \frac{2\alpha\hat{\Delta}_i}{(1 - \beta)\hat{\Delta}|\hat{V}_i|} + \frac{2}{\beta|\hat{V}_i|}\right) = \frac{3\alpha}{1 - \beta} + \frac{2k'}{\beta}.$$

By simple calculus, this bound is minimized at $\beta = \frac{1}{1 + \sqrt{\frac{3\alpha}{2k'}}} < 1$. □

4.3.4 Coresets by Sensitivity Sampling

Here, we apply the framework of Braverman et al. [46] as used in [108], using a slightly different range space. The ranges in the range space used to derive our result are open metric balls, which we now define.

Definition 4.3.4 *For $r \in \mathbb{R}_{\geq 0}$, $z \in Z$ and $Y \subseteq X$ we denote by $\mathrm{B}(z, r, Y) = \{y \in Y \mid \rho(y, z) < r\}$ the open metric ball with center z and radius r. We denote the set of all open metric balls by $\mathbb{B}(Y, Z) = \{\mathrm{B}(z, r, Y) \mid z \in Z, r \in \mathbb{R}_{\geq 0}\}$.*

Now, we are ready to analyze the computation of the actual ε-coresets. We use the reduction to uniform sampling, introduced by Feldman and Langberg [101] and improved by Braverman et al. [46], to apply Theorem 2.6.9. In the following, we adapt and modify the proof of Theorem 31 by Feldman et al. [108] and combine it with results by Munteanu et al. [206] to handle the involved scaling.

Theorem 4.3.5 *For $f \in F$ we let $\lambda(f) = \left\lceil |F| \cdot 2^{\lceil \log(\gamma(f)) \rceil} \right\rceil / |F|$, $\Lambda = \sum_{f \in F} \lambda(f)$, $\psi(f) = \frac{\lambda(f)}{\Lambda}$ and $\mathcal{D}$ be the VC dimension of the range space $(Y, \mathbb{B}(Y, Z))$.*

Let $\delta, \varepsilon \in (0, 1)$. A sample S of $\Theta(\varepsilon^{-2} \alpha k' (\mathcal{D} k \log(k) \log(\alpha k' n) \log(\alpha k') + \log(1/\delta)))$ elements $\tau_i \in T$, drawn independently with replacement with probability $\psi(f_i)$ and weighted by $w(f_i) = \frac{\Lambda}{|S| \lambda(f_i)}$ is a weighted ε-coreset for generalized k-median clustering of T with probability at least $1 - \delta$.

Proof. We define for $C \subseteq Z$ with $|C| = k$ the estimator

$$\widehat{\mathrm{cost}}(S, C) = \sum_{\tau_i \in S} w(f_i) \cdot \min_{c \in C} \rho(\tau_i, c) = \sum_{\tau_i \in S} w(f_i) \cdot f_i(C) = \sum_{\tau_i \in S} \frac{\Lambda}{|S| \lambda(f_i)} f_i(C)$$

for $\mathrm{cost}(T, C)$. We see that

$$\mathrm{Exp}\left[\widehat{\mathrm{cost}}(S, C)\right] = \sum_{i=1}^{|S|} \sum_{\tau_j \in T} \frac{\Lambda}{|S| \lambda(f_j)} f_j(C) \frac{\lambda(f_j)}{\Lambda} = \sum_{\tau_j \in T} f_j(C) = \mathrm{cost}(T, C),$$

thus $\widehat{\mathrm{cost}}(S, C)$ is unbiased. We want to bound the error of $\widehat{\mathrm{cost}}(S, C)$ by applying Theorem 2.6.9. To do so, we reduce the sensitivity sampling to uniform sampling as follows: We let G be a multiset that is a copy of F, where each $f \in F$ is contained $|F| \lambda(f)$ times and is scaled by $\frac{1}{|F| \lambda(f)}$. Note that $|G| = |F| \Lambda$. Also, $\psi(f) = \frac{|F| \lambda(f)}{|G|}$, $|F| \lambda(f)$ is integral for each $f \in F$ and

$$\sum_{g \in G} g(C) = \sum_{f \in F} \frac{|F| \lambda(f)}{|F| \lambda(f)} f(C) = \sum_{f \in F} f(C) = \mathrm{cost}(T, C).$$

Given a sample S', with $|S'| = |S|$, drawn independently and uniformly at random with replacement from G, for $C \subseteq Z$ with $|C| = k$ we define the estimator

$$\widetilde{\mathrm{cost}}(S', C) = \frac{|G|}{|S'|} \sum_{g \in S'} g(C)$$

for $\mathrm{cost}(T, C)$. We see that

$$\mathrm{Exp}\left[\widetilde{\mathrm{cost}}(S', C)\right] = \frac{|G|}{|S'|} \sum_{i=1}^{|S'|} \sum_{f \in F} \frac{f(C)}{|F| \lambda(f)} \frac{|F| \lambda(f)}{|G|} = \frac{1}{|S'|} \sum_{i=1}^{|S'|} \sum_{f \in F} f(C) = \sum_{g \in G} g(C).$$

Thus, $\widetilde{\mathrm{cost}}(S', C)$ is unbiased, too. We now assume that $S' = \left\{\frac{1}{|F|\lambda(f_i)} \cdot f_i \mid \tau_i \in S\right\}$. Then,

$$\widetilde{\mathrm{cost}}(S', C) = \frac{|G|}{|S'|} \sum_{g \in S'} g(C) = \frac{|F|\Lambda}{|S'|} \sum_{\tau_i \in S} \frac{1}{|F|\lambda(f_i)} f_i(C) = \sum_{\tau_i \in S} \frac{\Lambda}{|S|\lambda(f_i)} f_i(C) = \widehat{\mathrm{cost}}(S, C),$$

so the error bound for $\widetilde{\mathrm{cost}}(S', C)$, that we derive in the following, also applies to $\widehat{\mathrm{cost}}(S, C)$, hence S together with w is a weighted ε-coreset (see Problem 4.3.1). We now apply Theorem 2.6.9 with the given δ, $\varepsilon/2$ and $\eta = 1/\Lambda$, so the overall error is at most $\varepsilon \cdot \mathrm{cost}(T, C)$ for each $C \subseteq Z$ with $|C| = k$.

For $H \subseteq G$, $C \subseteq Z$ and $r \in \mathbb{R}_{\geq 0}$, we let $\mathrm{range}(H, C, r) = \{g \in H \mid g(C) \geq r\}$. Now, we let $(G, \mathcal{R})$ be a range space over G, where $\mathcal{R} = \{\mathrm{range}(G, C, r) \mid r \in \mathbb{R}_{\geq 0}, C \subseteq Z, |C| = k\}$. For all $C \subseteq Z$ with $|C| = k$ and all $H \subseteq G$ we have that

$$\begin{aligned}\sum_{g \in H} g(C) &= \sum_{g \in H} \int_0^\infty \mathbb{1}(g(C) \geq r) \;\mathrm{d}r = \int_0^\infty \sum_{g \in H} \mathbb{1}(g(C) \geq r) \;\mathrm{d}r \\ &= \int_0^\infty |\mathrm{range}(H, C, r)| \;\mathrm{d}r. \end{aligned} \tag{I}$$

Note that the indicator function is integrable under these circumstances and $|\mathrm{range}(H, C, r)|$ is a step function and is integrable, too. Using this identity, for all $C \subseteq Z$ with $|C| = k$ we now bound the error introduced by $\widehat{\mathrm{cost}}(S, C)$:

$$\begin{aligned}\left|\mathrm{cost}(T, C) - \widehat{\mathrm{cost}}(S, C)\right| = \left|\mathrm{cost}(T, C) - \widetilde{\mathrm{cost}}(S', C)\right| &= \left|\sum_{g \in G} g(C) - \frac{|G|}{|S'|} \sum_{g \in S'} g(C)\right| \\ &= \left|\int_0^\infty |\mathrm{range}(G, C, r)| \;\mathrm{d}r - \frac{|G|}{|S'|} \int_0^\infty |\mathrm{range}(S', C, r)| \;\mathrm{d}r\right| \\ &= \left|\int_0^\infty |\mathrm{range}(G, C, r)| - \frac{|G|}{|S'|} |\mathrm{range}(S', C, r)| \;\mathrm{d}r\right| \\ &\leq \int_0^\infty \left||\mathrm{range}(G, C, r)| - \frac{|G|}{|S'|} |\mathrm{range}(S', C, r)|\right| \mathrm{d}r.\end{aligned}$$

Here the second equation follows from Eq. (I).

In the following, let $\mathrm{rerror}(C, r) = \left||\mathrm{range}(G, C, r)| - \frac{|G|}{|S'|}|\mathrm{range}(S', C, r)|\right|$, $r_u(C) = \max_{g \in G} g(C)$, $R_1(C) = \{r \in \mathbb{R}_{\geq 0} \mid |\mathrm{range}(G, C, r)| \geq \eta \cdot |G|\}$ and $R_2(C) = \mathbb{R}_{\geq 0} \setminus R_1(C)$. Note that $R_1(C)$ and $R_2(C)$ are intervals due to the monotonicity of $|\mathrm{range}(G, C, r)|$. Furthermore, for $r \in (r_u(C), \infty)$ it holds that $|\mathrm{range}(G, C, r)| = 0$. Using these facts, we further derive:

$$\begin{aligned}\int_0^\infty \mathrm{rerror}(C, r) \;\mathrm{d}r &= \int_{R_1} \mathrm{rerror}(C, r) \;\mathrm{d}r + \int_{R_2} \mathrm{rerror}(C, r) \;\mathrm{d}r \\ &\leq \int_{R_1} \frac{\varepsilon}{2} |\mathrm{range}(G, C, r)| \;\mathrm{d}r + \int_{R_2} \frac{\varepsilon}{2} \eta |G| \;\mathrm{d}r \\ &\leq \frac{\varepsilon}{2} \int\limits_0^\infty |\mathrm{range}(G, C, r)| \;\mathrm{d}r + \frac{\varepsilon\eta|G|}{2} \int\limits_0^{r_u(C)} \;\mathrm{d}r \\ &= \frac{\varepsilon}{2} \sum_{g \in G} g(C) + \frac{\varepsilon\eta|G| r_u(C)}{2}.\end{aligned} \tag{III}$$

Here, the first inequality follows from Definition 2.6.8 and in the last equation we use Eq. (I). Finally, we bound the last summand in Eq. (III). First note that we have for each $g \in G$:

$$\frac{g(C)}{\sum_{h\in G} h(C)} = \frac{\frac{1}{|F|\lambda(f)} f(C)}{\sum_{h\in F} h(C)} \le \frac{\lambda(f)}{|F|\lambda(f)} \iff \frac{g(C)}{\sum_{h\in G} h(C)} \le \frac{1}{|F|},$$

where $f \in F$ is the function that g is a copy of and the inequality follows from Definition 4.3.2. This implies $r_u(C) \le \frac{1}{|F|}\sum_{h\in G} h(C)$. We now further derive:

$$\frac{\varepsilon\eta|G|r_u(C)}{2} \le \frac{\varepsilon}{2}\frac{1}{\Lambda}|F|\Lambda\frac{1}{|F|}\sum_{g\in G} g(C) = \frac{\varepsilon}{2}\sum_{g\in G} g(C).$$

So, all in all $|\mathrm{cost}(T,C) - \widehat{\mathrm{cost}}(S,C)| \le \varepsilon \cdot \mathrm{cost}(T,C)$ for all $C \subseteq Z$ with $|C| = k$.

The claim now follows from the facts that

- $\gamma(f) \le \lambda(f) \le 2\cdot\gamma(f) + \frac{1}{|F|}$ for each $f \in F$, thus $\Lambda \le 2\cdot\Gamma(F) + 1$ and $\Gamma(F) \in O(\alpha k')$ by Lemma 4.3.3 and
- $(G, \mathcal{R})$ has VC dimension in $O(\mathcal{D}k\log(k)\log(\alpha k' n))$ by the following Lemma 4.3.6.

□

Lemma 4.3.6 *$(G,\mathcal{R})$ has VC dimension $O(\mathcal{D}k\log(k)\log(\alpha k' n))$, where $\mathcal{D}$ is the VC dimension of the range space $(Y, \mathbb{B}(Y,Z))$.*

Proof. First, we assume that there exists a $\varphi \in \mathbb{R}$, such that $\lambda(f) = \varphi$ for all $f \in F$. Then all functions in G are scaled uniformly and we can completely neglect the scaling. In this case $(G,\mathcal{R})$ has equal VC dimension as

$$Q_1 = (T, \{T \setminus (\mathrm{B}(c_1,r,T) \cup \dots \cup \mathrm{B}(c_k,r,T)) \mid \{c_1,\dots,c_k\} \subseteq Z, r \in \mathbb{R}_{\ge 0}\}),$$

the VC dimension of Q_1 is at most the VC dimension of

$$Q_2 = (T, \{T \setminus (B_1 \cup \dots \cup B_k) \mid B_1,\dots,B_k \in \mathbb{B}(T,Z)\}),$$

Q_2 is the projection (see Definition 2.6.2) of

$$Q_3 = (Y, \{Y \setminus (B_1 \cup \dots \cup B_k) \mid B_1,\dots,B_k \in \mathbb{B}(Y,Z)\})$$

onto T and has thus at most the VC dimension of Q_3 and finally, Q_3 is the complementary range space (see Definition 2.6.3) of

$$Q_4 = (Y, \{B_1 \cup \dots \cup B_k \mid B_1,\dots,B_k \in \mathbb{B}(Y,Z)\})$$

and has thus equal VC dimension as Q_4. By the k-fold union theorem [39, Lemma 2.3.2] Q_4 has VC dimension $O(\mathcal{D}\cdot k\log(k))$, where $\mathcal{D}$ is the VC dimension of $(Y,\mathbb{B}(Y,Z))$. For the following, let c be the constant hidden in this O-notation.

Contrary to the former case, if there are $t > 1$ distinct values $\Phi = \{\varphi_1,\dots,\varphi_t\} \subset \mathbb{R}$, such that $\lambda(f) \in \Phi$ for each $f \in F$ and $\forall i \in [t] \exists f \in F : \lambda(f) = \varphi_i$, we apply the techniques of Munteanu et al. [206] (see Lemma 11 and Theorem 15 therein).

First, assume that the VC dimension of $(G, \mathcal{R})$ is greater than $t \cdot c \cdot \mathcal{D} \cdot k \log(k)$. Hence, there exists a set $G' \subseteq G$ with $|G'| > t \cdot c \cdot \mathcal{D} \cdot k \log(k)$, such that $|\{G' \cap R \mid R \in \mathcal{R}\}| = 2^{|G'|}$. Let $\{G_1, \dots, G_t\}$ be a partition of G, such that for each $g \in G_i$ we have $g = \frac{1}{|F|\lambda(f)} f = \frac{1}{|F|\varphi_i} f$ for an $f \in F$. Furthermore, for $i \in [t]$, let $G'_i = G' \cap G_i$.

By disjointness, we have $|\{G'_i \cap R_i \mid R_i = (R \cap G_i), R \in \mathcal{R}\}| = 2^{|G'_i|}$ for each $i \in [t]$ and also there must exist at least one $j \in [t]$, such that $|G'_j| \geq \frac{|G'|}{t} > \frac{t \cdot c \cdot \mathcal{D} \cdot k \log(k)}{t} = c \cdot \mathcal{D} \cdot k \log(k)$, hence the projection of $(G, \mathcal{R})$ on G_j has VC dimension greater than $c \cdot \mathcal{D} \cdot k \log(k)$. This is a contradiction to the former case of uniformly scaled functions in G, thus $(G, \mathcal{R})$ has VC dimension $O(t \cdot \mathcal{D} \cdot k \log(k))$ in this case.

Now, we bound t. Recall that for each $f \in F$, $\lambda(f) = \left\lceil |F| 2^{\lceil \log(\gamma(f)) \rceil} \right\rceil / |F|$. Furthermore for each $i \in [k']$ and $\tau_j \in \hat{V}_i$ (see Lemma 4.3.3),

$$\left(1 + \sqrt{\frac{3\alpha}{2k'}}\right) \frac{2}{|\hat{V}_i|} \leq \gamma(f_j) \leq \left(1 + \sqrt{\frac{2k'}{3\alpha}}\right) \left(\alpha + \frac{2\alpha}{|\hat{V}_i|}\right) + \left(1 + \sqrt{\frac{3\alpha}{2k'}}\right) \frac{2}{|\hat{V}_i|}.$$

Therefore, there can be at most

$$\begin{aligned} &\log \left(\frac{\left(1 + \sqrt{\frac{2k'}{3\alpha}}\right) \left(\alpha + \frac{2\alpha}{|\hat{V}_i|}\right) + \left(1 + \sqrt{\frac{3\alpha}{2k'}}\right) \frac{2}{|\hat{V}_i|}}{\left(1 + \sqrt{\frac{3\alpha}{2k'}}\right) \frac{2}{|\hat{V}_i|}} \right) \\ \leq &\log \left(\frac{\left(1 + \sqrt{\frac{2k'}{3\alpha}}\right)}{\left(1 + \sqrt{\frac{3\alpha}{2k'}}\right)} (\alpha n/2 + \alpha) + 1 \right) \\ \leq &\log(9\alpha k'(\alpha n/2 + \alpha) + 1) \end{aligned}$$

distinct values of $2^{\lceil \log(\gamma(f)) \rceil}$, which upper bounds the number of distinct values of $\lambda(f)$.

We conclude that the VC dimension of $(G, \mathcal{R})$ is in $O(\mathcal{D} k \log(k) \log(\alpha k' n))$. □

4.3.5 Coresets for (k, ℓ)-Median Clustering under the Fréchet Distance

Here we present an algorithm for computing ε-coresets for (k, ℓ)-median clustering of polygonal curves under the Fréchet distance. The algorithm builds upon our results from Sections 4.3.3 and 4.3.4. To apply those, we still need a bound on the VC dimension of metric balls under the Fréchet distance, which we derive in the following theorem.

Theorem 4.3.7 *The VC dimension of* $(\mathbb{R}^d_m, \mathbb{B}(\mathbb{R}^d_m, \mathbb{R}^d_\ell))$ *is* $O\left(\ell^2 \log(\ell m)\right)$.

Proof. We argue that the claim follows from Theorem 18 by Driemel et al. [88]. First, in their paper polygonal curves do not need to adhere the restriction that no three consecutive vertices may be collinear and they define $\mathbb{R}^d_m$ to be the polygonal curves of exactly m vertices. However, our definitions match by simulating the addition of collinear vertices to those curves in $\mathbb{R}^d_m$ with less than m vertices.

Now, looking into their proof, we can slightly modify the Fréchet distance predicates to use "$<$" instead of "$\leq$", thereby altering the geometric primitives by letting $B_r(p) = \{x \in \mathbb{R}^d \mid \|x-p\| < r\}$, $D_r(\overline{st}) = \{x \in \mathbb{R}^d \mid \exists p \in \overline{st} : \|p-x\| < r\}$, $C_r(\overline{st}) = \{x \in \mathbb{R}^d \mid \exists p \in \ell(\overline{st}) : \|p-x\| < r\}$, $R_r(\overline{st}) = \{p+u \mid p \in \overline{st}, u \in \mathbb{R}^d, \langle t-s, u\rangle = 0, \|u\| < r\}$ and $M_r(\overline{st})$ satisfy $\|p_1 - q_1\| < r$ and $\|p_2 - q_2\| < r$, which does not affect the remainder of the proof and thus yields the same bound on the VC dimension. □

To compute ε-coresets for (k,ℓ)-median clustering under the Fréchet distance, we first need to compute bounds on the sensitivities. For $k > 1$ we use Algorithm 13, a modification of [87, Algorithm 3], which we now present.

Algorithm 13 Constant Factor Approximation for (k,ℓ)-Median Clustering

```
procedure (k,ℓ)-MEDIAN-96-APPROXIMATION($T = \{\tau_1, \dots, \tau_n\}$)
    for $i = 1, \dots, n$ do
        $\hat{\tau}_i \leftarrow$ approximate minimum-error $\ell$-simplification of $\tau_i$        ▷ [21, 149]
    $C \leftarrow$ Chen's algorithm with $\varepsilon = 0.5, \lambda = \delta$ on $\{\hat{\tau}_1, \dots, \hat{\tau}_n\}$        ▷ [72, Theorem 6.2]
    return $C$
```

We prove the correctness and analyze the running time of Algorithm 13.

Theorem 4.3.8 *Given a parameter $\delta \in (0,1)$ and a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, Algorithm 13 returns with probability at least $1-\delta$ a 109-approximate (k,ℓ)-median solution for T in time $O(nm\log(1/\delta)\log(m) + nm^3\log m)$.*

Proof. We assume that the approximate minimum-error ℓ-simplifications are computed combining the algorithms by Alt and Godau [21] and Imai and Iri [149], so the approximation factor is 4 by Theorem 2.8.6. For $i \in [n]$, let $\hat{\tau}_i$ be the simplification of τ_i and let $\hat{T} = \{\hat{\tau}_1, \dots, \hat{\tau}_n\}$. Note that $\mathrm{d_F}(\tau_i, \hat{\tau}_i) \leq 4 \cdot \mathrm{d_F}(\tau, \sigma)$ for all $i \in [n]$ and $\sigma \in \mathbb{R}^d_\ell$.

Let $\hat{C}^* = \{\hat{c}^*_1, \dots, \hat{c}^*_k\} \subset \mathbb{R}^d_\ell$ be an optimal (k,ℓ)-median solution for $\hat{T}$ and let $C' = \{c'_1, \dots, c'_k\} \subseteq \hat{T}$ be an optimal solution to the discrete (k,ℓ)-median problem for $\hat{T}$, i.e. the centers are chosen among the input. Breaking ties arbitrarily, we assume that every $\hat{\tau} \in \hat{T}$ has a unique nearest neighbor in $\hat{C}^*$ and for $i \in [k]$ we define $\hat{T}_i = \{\hat{\tau} \in \hat{T} \mid \forall j \in [k] : \mathrm{d_F}(\hat{\tau}, \hat{c}^*_i) \leq \mathrm{d_F}(\hat{\tau}, \hat{c}^*_j)\}$, such that $\hat{T}_1, \dots, \hat{T}_k$ form a partition of $\hat{T}$. By the triangle inequality:

$$\begin{aligned}
\mathrm{cost}\left(\hat{T}, C'\right) &= \min_{\substack{C \subseteq \hat{T}\\ |C|=k}} \sum_{i=1}^{k} \sum_{\hat{\tau} \in \hat{T}_i} \min_{c \in C} \mathrm{d_F}(\hat{\tau}, c) \leq \min_{\substack{C \subseteq \hat{T}\\ |C|=k}} \sum_{i=1}^{k} \sum_{\hat{\tau} \in \hat{T}_i} \left(\mathrm{d_F}(\hat{\tau}, \hat{c}^*_i) + \min_{c \in C} \mathrm{d_F}(\hat{c}^*_i, c)\right) \\
&= \mathrm{cost}\left(\hat{T}, \hat{C}^*\right) + \min_{\substack{C \subseteq \hat{T}\\ |C|=k}} \sum_{i=1}^{k} \sum_{\hat{\tau} \in \hat{T}_i} \min_{c \in C} \mathrm{d_F}(\hat{c}^*_i, c) \\
&= \mathrm{cost}\left(\hat{T}, \hat{C}^*\right) + \sum_{i=1}^{k} \sum_{\hat{\tau} \in \hat{T}_i} \min_{\hat{\sigma} \in \hat{T}_i} \mathrm{d_F}(\hat{c}^*_i, \hat{\sigma}).
\end{aligned}$$

For each $i \in [k]$ there must exist a $\hat{\sigma} \in \hat{T}_i$ with $\mathrm{d_F}(\hat{\sigma}, \hat{c}^*_i) \leq \sum_{\hat{\tau} \in \hat{T}_i} \mathrm{d_F}(\hat{\tau}, \hat{c}^*_i)/|\hat{T}_i|$, since otherwise $\sum_{i=1}^{k} \sum_{\hat{\tau} \in \hat{T}_i} \min_{\hat{\sigma} \in \hat{T}_i} \mathrm{d_F}(\hat{c}^*_i, \hat{\sigma}) > \mathrm{cost}\left(\hat{T}, \hat{C}^*\right)$, which is a contradiction. We conclude that $\mathrm{cost}\left(\hat{T}, C'\right) \leq 2\,\mathrm{cost}\left(\hat{T}, \hat{C}^*\right)$. Also, by [72, Theorem 6.2] $\mathrm{cost}\left(\hat{T}, C\right) \leq 10.5\,\mathrm{cost}\left(\hat{T}, C'\right)$.

Now, let $C^* = \{c_1^*, \ldots, c_k^*\} \subset \mathbb{R}^d_\ell$ be an optimal (k,ℓ)-median solution for T and $C = \{c_1, \ldots, c_k\}$ be a solution returned by Algorithm 13 for $\hat{T}$. We derive:

$$\begin{aligned}\operatorname{cost}(T,C) &\leq \sum_{i=1}^{n}(\mathrm{d_F}(\tau_i,\hat{\tau}_i) + \min_{j\in[k]} \mathrm{d_F}(\hat{\tau}_i, c_j)) = \sum_{i=1}^{n} \mathrm{d_F}(\tau_i,\hat{\tau}_i) + \operatorname{cost}\left(\hat{T}, C\right) \\ &\leq 4\operatorname{cost}(T,C^*) + 21\operatorname{cost}\left(\hat{T},\hat{C}^*\right) \leq 4\operatorname{cost}(T,C^*) + 21\operatorname{cost}\left(\hat{T},C^*\right) \\ &\leq 4\operatorname{cost}(T,C^*) + 21\left(\sum_{i=1}^{n}(\mathrm{d_F}(\hat{\tau}_i,\tau_i) + \min_{j\in[k]}\mathrm{d_F}(\tau_i,c_j^*))\right) \\ &\leq 4\operatorname{cost}(T,C^*) + 84\operatorname{cost}(T,C^*) + 21\operatorname{cost}(T,C^*) \leq 109\operatorname{cost}(T,C^*).\end{aligned}$$

We now analyze the running time. Computing the simplifications takes time $O(nm^3\log m)$, see Theorem 2.8.6. Further, we incorporate the given probability of failure (see [72, Theorem 3.6]) into the running time stated in [72, Theorem 6.2]. Hence, Chen's algorithm can be run in time $O(nm\log(1/\delta)\log m)$ when the distances are computed using Alt and Godau's algorithm [21]. □

For $k = 1$, we use the more efficient Algorithm 3 to compute the bounds on the sensitivities. We now present the algorithm for computing weighted ε-coresets for (k,ℓ)-median clustering.

Algorithm 14 Coresets for (k,ℓ)-Median Clustering

```
procedure $(k,\ell)$-MEDIAN-CORESET($T = \{\tau_1,\ldots,\tau_n\}, \delta, \varepsilon$)
    if $k = 1$ then
        $\hat{c} \leftarrow \ell$-Median-34-Approximation($T, \delta/2$) (Algorithm 3)
        $\hat{C} = \{\hat{c}\}$
    else
        $\hat{C} = \{\hat{c}_1,\ldots,\hat{c}_k\} \leftarrow (k,\ell)$-Median-96-Approximation($T, \delta/2$) (Algorithm 13)
    compute $\hat{V}_1,\ldots,\hat{V}_k$, $\hat{\Delta}_1,\ldots,\hat{\Delta}_k$ and $\gamma$ w.r.t. $\hat{C}$ (cf. Lemma 4.3.3)
    compute $\lambda$, $\Lambda$ w.r.t. $\gamma$ and $\psi$ w.r.t. $\lambda$ (cf. Theorem 4.3.5)
    $S \leftarrow$ sample $\Theta(k\varepsilon^{-2}(d^2\ell^2 k\log(d\ell m)\log(kn)\log^2(k) + \log(1/(2\delta))))$ elements from $T$
            independently with replacement with respect to $\psi$
    compute $w$ w.r.t. $\lambda$, $\Lambda$ and $S$ (cf. Theorem 4.3.5)
    return $S$ and $w$
```

We prove the correctness and analyze the running time of Algorithm 14. Also, we analyze the size of the resulting ε-coreset.

Theorem 4.3.9 *Given a set $T = \{\tau_1,\ldots,\tau_n\} \subset \mathbb{R}^d_m$ and parameters $\delta,\varepsilon \in (0,1)$, Algorithm 14 computes a weighted ε-coreset of size $O(\varepsilon^{-2}(\log(m)\log(n) + \log(1/\delta)))$ for (k,ℓ)-median clustering with probability at least $1-\delta$, in time*

$$O(nm\log(m)\log(1/\delta) + nm^3\log m + \varepsilon^{-2}(\log(m)\log(n) + \log 1/\delta))$$

for $k > 1$ and

$$O(nm\log(m) + m^2\log(m)\log^2(1/\delta) + m^3\log m + \varepsilon^{-2}(\log(m)\log(n) + \log 1/\delta))$$

for $k = 1$.

Proof. First note that by a union bound, with probability at least $1-\delta$ Algorithm 13, respectively Algorithm 3, and the sampling are simultaneously successful (see Theorems 4.3.5 and 4.3.8 and Corollary 3.1.18). The correctness of the algorithm follows from the observations that the (k,ℓ)-median clustering objective fits the generalized k-median clustering objective with $X = \mathbb{R}^d_*$, $Y = \mathbb{R}^d_m \subset X$ and $Z = \mathbb{R}^d_\ell \subset X$, therefore Lemma 4.3.3 and Theorem 4.3.5 can be applied, and the VC dimension of $(X, \mathbb{B}(Y,Z))$ is $O(\ell^2 \log(\ell m))$ by Theorem 4.3.7.

We now analyze the running time. $\hat{V}_1, \dots, \hat{V}_k$, $\hat{\Delta}_1, \dots, \hat{\Delta}_k$ and γ can be computed in time $O(nm\log(m))$ using Alt and Godau's algorithm [21]. λ, Λ and ψ can be computed in time[2] $O(n)$ and the sampling can be carried out in time $O(\varepsilon^{-2}(\log(m)\log(n) + \log 1/\delta))$. Finally, w can be computed in time $O(n)$. If $k > 1$ we run Algorithm 13 in time $O(nm\log(1/\delta)\log(m) + nm^3\log(m))$, see Theorem 4.3.8. Else, we run Algorithm 3 in time $O(m^2\log(m)\log^2(1/\delta) + m^3\log m)$, see Corollary 3.1.18.

All in all, the running time is then

$$O(nm\log(1/\delta)\log(m) + nm^3\log m + \varepsilon^{-2}(\log(m)\log(n) + \log(1/\delta)))$$

for $k > 1$ and

$$O(nm\log(m) + m^2\log(m)\log^2(1/\delta) + m^3\log m + \varepsilon^{-2}(\log(m) + \log(1/\delta)))$$

for $k = 1$. □

Finally, we note that our results also imply the existence of ε-coresets for (k,ℓ)-median clustering under the discrete and weak[3] Fréchet and the (discrete and continuous) Hausdorff distances, as the VC dimension of the range spaces induced by the metric balls under these measures is similarly bounded [88].

Application to the ℓ-Median Problem

Here, we present a modification of Algorithm 4 for approximating Problem 3.1.1. Our modification uses ε-coresets to improve the running time of the algorithm, rendering it tractable in a big data setting. The algorithm uses ε-coresets every time it has to evaluate the cost of a center set. The dramatic effect of this small modification is that we nearly lose the original linear running time dependency on n in the most time-consuming part of the algorithm, rendering it practical in the setting of big data, where we have a lot of curves of much smaller complexity than number $(\ell < m \ll n)$.

We note that we can not use ε-coresets to improve the combination of Algorithm 9 and Algorithm 4, respectively Algorithm 9 and Algorithm 5. This is due to the fact that Algorithm 9 calls itself with subsets of the original input as input. Therefore, we would need a coreset for every possible subset, which is prohibitive even in asymptotic running time.

[2] Here we note that from the bounds in Lemma 4.3.6 it can be seen that the integers resulting from the application of the ceil function require only $O(\log n)$ bits.

[3] A variant of the continuous Fréchet distance that, informally speaking, allows a point on one curve to be matched to multiple points on the second curve.

Algorithm 15 ℓ-Median by Simple Shortcutting and ε-Coreset

```
procedure ℓ-MEDIAN-(5 + ε)-APPROXIMATION($T = \{\tau_1, \dots, \tau_n\}, \delta, \varepsilon$)
    $\hat{c} \leftarrow \ell$-Median-34-Approximation($T, \delta/4$) (Algorithm 3)
    $\varepsilon' \leftarrow \varepsilon/67$, $P \leftarrow \emptyset$
    $(T', w) \leftarrow (1, 2\ell-2)$-Median-Coreset($T, \delta/4, \varepsilon'$)
    $\Delta \leftarrow \mathrm{cost}_w(T', \{\hat{c}\})$, $\Delta_u \leftarrow \Delta/(1-\varepsilon')$, $\Delta_l \leftarrow \Delta/((1+\varepsilon')34)$
    $S \leftarrow$ sample $\lceil -2(\varepsilon')^{-1}(\ln(\delta) - \ln(4)) \rceil$ curves from $T$ uniformly
        and independently with replacement
    $W \leftarrow$ sample $\lceil -64(\varepsilon')^{-2}(\ln(\delta) - \ln(\lceil -8(\varepsilon')^{-1}(\ln(\delta) - \ln(4)) \rceil)) \rceil$ curves
        from $T$ uniformly and independently with replacement
    $c \leftarrow$ arbitrary element from $\arg\min_{s \in S} \mathrm{cost}(W, s)$
    for $i = 1, \dots, |c|$ do
        $P \leftarrow P \cup \mathbb{G}(B(v_i^c, (3 + 4\varepsilon')\Delta_u/n), \varepsilon'\Delta_l/(n\sqrt{d}))$        ▷ $v_i^c$: $i^{\text{th}}$ vertex of $c$
    $C \leftarrow$ set of all polygonal curves with $2\ell - 2$ vertices from $P$
    return $\arg\min_{c' \in C} \mathrm{cost}_w(T', \{c'\})$
```

We prove the correctness and analyze the running time of Algorithm 15.

Theorem 4.3.10 *Given two parameters $\delta \in (0,1)$, $\varepsilon \in (0, 1/2]$ and a set $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_m$ of polygonal curves, with probability at least $1 - \delta$ Algorithm 15 returns a $(5 + \varepsilon)$-approximate ℓ-median for T with $2\ell - 2$ vertices, in time*

$$O\left(nm \log m + m^2 \log(m) \log^2(1/\delta) + m^{2\ell-1} \varepsilon^{-2\ell d + 2d - 2} (\log(m) \log(n) + \log(1/\delta)) \log(m)\right).$$

Proof. Let $c^* \in \arg\min_{c \in \mathbb{R}^d_\ell} \mathrm{cost}(T, c)$ be an optimal ℓ-median for T. The expected distance between $s \in S$ and c^* is

$$\mathrm{Exp}[\mathrm{d_F}(s, c^*)] = \sum_{i=1}^{n} \mathrm{d_F}(\tau_i, c^*) \cdot \frac{1}{n} = \frac{\mathrm{cost}(T, c^*)}{n}.$$

Now, using Markov's inequality, for every $s \in S$ we have

$$\Pr[\mathrm{d_F}(s, c^*) > (1 + \varepsilon) \mathrm{cost}(T, c^*)/n] \leq \frac{\mathrm{cost}(T, c^*) n^{-1}}{(1 + \varepsilon) \mathrm{cost}(T, c^*) n^{-1}} = \frac{1}{1 + \varepsilon},$$

therefore by independence

$$\Pr\left[\min_{s \in S} \mathrm{d_F}(s, c^*) > (1 + \varepsilon) \mathrm{cost}(T, c^*)/n\right] \leq \frac{1}{(1 + \varepsilon)^{|S|}} \leq \exp\left(-\frac{\varepsilon |S|}{2}\right).$$

Hence, with probability at most $\exp\left(-\frac{\varepsilon \left\lceil -\frac{2(\ln(\delta) - \ln(4))}{\varepsilon} \right\rceil}{2}\right) \leq \delta/4$ there is no $s \in S$ with $\mathrm{d_F}(s, c^*) \leq (1 + \varepsilon) \frac{\mathrm{cost}(T, c^*)}{n}$. Now, assume there is a $s \in S$ with $\mathrm{d_F}(s, c^*) \leq (1 + \varepsilon) \mathrm{cost}(T, c^*)/n$. We do not want any $t \in S \setminus \{s\}$ with $\mathrm{cost}(T, t) > (1 + \varepsilon) \mathrm{cost}(T, s)$ to have $\mathrm{cost}(W, t) \leq \mathrm{cost}(W, s)$. Using Theorem 2.4.19, we conclude that this happens with probability at most

$$\begin{aligned} \exp\left(-\frac{\varepsilon^2 \lceil -64\varepsilon^{-2}(\ln(\delta) - \ln(\lceil -8(\varepsilon')^{-1}(\ln(\delta) - \ln(4)) \rceil)) \rceil}{64}\right) &\leq \frac{\delta}{\lceil -8(\varepsilon')^{-1}(\ln(\delta) - \ln(4)) \rceil} \\ &\leq \frac{\delta}{4|S|}, \end{aligned}$$

for each $t \in S \setminus \{s\}$. Also, with probability at most $\delta/4$ Algorithm 3 fails to compute a 34-approximate ℓ-median $\hat{c} \in \mathbb{R}^d_\ell$ for T, see Corollary 3.1.18, and with probability at most $\delta/4$, Algorithm 14 fails to compute a weighted ε-coreset for T, see Theorem 4.3.5.

Using a union bound over these bad events, we conclude that with probability at least $1-\delta$,

- Algorithm 15 samples a curve $s \in S$ with $\mathrm{d_F}(s, c^*) \leq (1+\varepsilon)\,\mathrm{cost}(T, c^*)/n$,
- Algorithm 15 assigns c to a curve $t \in S$ with $\mathrm{cost}(T,t) \leq (1+\varepsilon)\,\mathrm{cost}(T,s)$,
- Algorithm 3 computes a 34-approximate ℓ-median $\hat{c} \in \mathbb{R}^d_\ell$ for T, i.e., $\mathrm{cost}(T,c^*) \leq \mathrm{cost}(T,\hat{c}) \leq 34\,\mathrm{cost}(T,c^*)$
- and Algorithm 14 computes a weighted ε-coreset for T.

Using the triangle inequality yields

$$\begin{aligned}\sum_{\tau\in T}(\mathrm{d_F}(t,c^*) - \mathrm{d_F}(\tau,c^*)) \leq \sum_{\tau\in T}\mathrm{d_F}(t,\tau) &\leq (1+\varepsilon)\sum_{\tau\in T}\mathrm{d_F}(s,\tau) \\ &\leq (1+\varepsilon)\sum_{\tau\in T}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,s)),\end{aligned}$$

which is equivalent to

$$\begin{aligned}& n\cdot \mathrm{d_F}(t,c^*) \leq (2+\varepsilon)\,\mathrm{cost}(T,c^*) + (1+\varepsilon)n(1+\varepsilon)\,\mathrm{cost}(T,c^*)/n \\ \iff\ & \mathrm{d_F}(t,c^*) \leq (3+4\varepsilon)\,\mathrm{cost}(T,c^*)/n.\end{aligned}$$

Let $v_1^t,\dots,v_{|t|}^t$ be the vertices of t. By Lemma 3.1.19 there exists a polygonal curve $c' \in \mathbb{R}^d_{2\ell-2}$ with every vertex contained in one of $B(v_1^t, \mathrm{d_F}(c^*,t)),\dots,B(v_{|t|}^t, \mathrm{d_F}(c^*,t))$ and $\mathrm{d_F}(t,c') \leq \mathrm{d_F}(t,c^*)$. We have $\mathrm{d_F}(t,c') \leq \mathrm{d_F}(t,c^*) \leq (3+4\varepsilon)\,\mathrm{cost}(T,c^*)/n$. Furthermore, by the ε-coreset gurarantee, see Problem 4.3.1, we have $|\Delta - \mathrm{cost}(T,\hat{c})| \leq \varepsilon\,\mathrm{cost}(T,\hat{c})$. Therefore, $\Delta_l = \Delta/(34(1+\varepsilon)) \leq \mathrm{cost}(T,c^*) \leq \Delta_u = \Delta/(1-\varepsilon)$ and $\mathrm{d_F}(t,c') \leq (3+4\varepsilon)\Delta_u/n$. We conclude that the set C of all curves with up to $2\ell-2$ vertices from P, the union of the grid covers, contains a curve $c'' \in \mathbb{R}^d_{2\ell-2}$ with distance at most $\frac{\varepsilon\Delta_l}{n} \leq \varepsilon\frac{\mathrm{cost}(T,c^*)}{n}$ between every corresponding pair of vertices of c' and c'', thus $\mathrm{d_F}(t,c'') \leq (3+5\varepsilon)\,\mathrm{cost}(T,c^*)/n$.

In the last step, Algorithm 15 returns a curve $\tilde{c} \in C$, that evaluates best against the ε-coreset. By the ε-coreset guarantee and the range of ε, we know that $\mathrm{cost}(T,\tilde{c}) \leq (1+\varepsilon)/(1-\varepsilon)\,\mathrm{cost}(T,c'') \leq (1+4\varepsilon)\,\mathrm{cost}(T,c'')$. We can now bound the cost of $\tilde{c}$ as follows:

$$\begin{aligned}\mathrm{cost}(T,\tilde{c}) &\leq (1+4\varepsilon)\sum_{\tau\in T}\mathrm{d_F}(\tau,c'') \leq (1+4\varepsilon)\sum_{\tau\in T}(\mathrm{d_F}(\tau,t) + \mathrm{d_F}(t,c'')) \\ &\leq (1+4\varepsilon)\,\mathrm{cost}(T,t) + (1+4\varepsilon)(3+5\varepsilon)\,\mathrm{cost}(T,c^*) \\ &\leq (1+\varepsilon)(1+4\varepsilon)\,\mathrm{cost}(T,s) + (3+37\varepsilon)\,\mathrm{cost}(T,c^*) \\ &\leq (1+9\varepsilon)\sum_{\tau\in T}(\mathrm{d_F}(\tau,c^*) + \mathrm{d_F}(c^*,s)) + (3+37\varepsilon)\,\mathrm{cost}(T,c^*) \\ &\leq (4+48\varepsilon)\,\mathrm{cost}(T,c^*) + (1+\varepsilon)(1+9\varepsilon)\,\mathrm{cost}(T,c^*) \\ &\leq (5+67\varepsilon)\,\mathrm{cost}(T,c^*)\end{aligned}$$

Finally, we rescale ε by $\frac{1}{67}$ to obtain the desired approximation guarantee.

We now discuss the running time. Algorithm 3 has running time $O(m^2 \log(m) \log^2(1/\delta) + m^3 \log m)$, see Corollary 3.1.18 and Algorithm 14 has running time

$$O(nm \log(m) + m^2 \log(m) \log^2(1/\delta) + m^3 \log m + \varepsilon^{-2}(\log(m) \log(n) + \log(1/\delta))),$$

see Theorem 4.3.5. The ε-coreset has size $O(\varepsilon^{-2}(\log(m) \log(n) + \log(1/\delta)))$, therefore $\mathrm{cost}_w(T', \hat{c})$ can be evaluated in time $O(m\varepsilon^{-2} \log(m)(\log(m) \log(n) + \log(1/\delta)))$, using Alt and Godau's algorithm [21] to compute the distances.

The sample S has size $O\left(\frac{\ln(1/\delta)}{\varepsilon}\right)$ and the sample W has size $O\left(\frac{\ln(1/\delta)}{\varepsilon^2}\right)$. Evaluating each curve of S against W takes time $O\left(\frac{m^2 \log(m) \log^2(1/\delta)}{\varepsilon^3}\right)$, using Alt and Godau's algorithm [21] to compute the distances.

Now, c has up to m vertices and every grid consists of $\left(\frac{\frac{2(3+4\varepsilon')\Delta_u}{n}}{\frac{2\varepsilon'\Delta_l}{n\sqrt{d}}}\right)^d = \left(\frac{(3+4\varepsilon')\sqrt{d}}{\varepsilon'} 34(1+\varepsilon)\right)^d \in O\left(\frac{1}{\varepsilon^d}\right)$ points (note that $\Delta_u/\Delta_l = (1+\varepsilon')/(1-\varepsilon')34 \leq 34(1+\varepsilon)$). Therefore, we have $O\left(\frac{m}{\varepsilon^d}\right)$ points in P and Algorithm 15 enumerates all combinations of $2\ell - 2$ points from P taking time $O\left(\frac{m^{2\ell-2}}{\varepsilon^{(2\ell-2)d}}\right)$. Afterwards, these candidates are evaluated against the ε-coreset, which takes time

$$O\left(\frac{m^{2\ell-1}(\log(m) \log(n) + \log(1/\delta)) \log(m)}{\varepsilon^{2\ell d - 2d + 2}}\right),$$

using Alt and Godau's algorithm [21] to compute the distances. All in all, we then have running time

$$O\left(nm \log m + m^2 \log(m) \log^2(1/\delta) + \frac{m^{2\ell-1}(\log(m) \log(n) + \log(1/\delta)) \log(m)}{\varepsilon^{2\ell d - 2d + 2}}\right).$$

□

5 High Complexity and High Dimensionality

The number of input elements, their complexity and the number of dimensions of the ambient space are the three main parameters that determine the running times (and storage space consumption) of the algorithms considered so far, and we mainly focused on obtaining efficient computational methods for a large number of input elements, while not prioritizing the complexity and neglecting the ambient dimension – but algorithms running times are often exponential in this parameter. In this chapter we focus on reducing the dimension, respectively complexity, of polygonal curves in high-dimensional Euclidean space and polygonal curves of high complexity.

5.1 Dimension Reduction for Curves

Dimension reduction is a collective term for methods that reduce the dimension of a data set, which is usually a set $P \subset \mathbb{R}^d$ of points. Of course, there are two aspects in computing that motivate us to reduce the dimension of the input, namely space and running time, which both depend on the dimension. Arguably the most prominent examples of dimension reduction techniques are principal component analysis (PCA) and (metric) embeddings.

In PCA one aims to compute the principal components of P. These are unit vectors that form an orthogonal basis of $\mathbb{R}^d$ that best describes the data. To compute the principal components one first translates the data set such that the origin is the center of mass, i.e., $\sum_{p \in P} \frac{p}{|P|} = 0$. The first principal component c_1 is then the unit length vector w that maximizes the sum of squared lengths of the position vectors of the (orthogonal) projections onto the line through the origin that is supported by w, i.e., $v_1 = \sum_{p \in P} \langle w, p \rangle^2$. To compute the i^{th} principal component, for $i > 1$, one first computes a new data set $P_{i-1} = \{p - \sum_{j=1}^{i-1} \langle c_j, p \rangle \mid p \in P\}$, thereby removing the extent in the directions of the previous principal components from the original data set, and then picks as c_i the unit length vector w that maximizes $v_i = \sum_{p \in P_{i-1}} \langle w, p \rangle^2$. Now, let $X_P \in \mathbb{R}^{|P| \times d}$ be a matrix whose rows are the points from P. In fact, c_i is the eigenvector of $X = X_P^T X_P$, corresponding to the i^{th} largest eigenvalue, i.e., v_i, of X and since X is a positive semi-definite symmetric matrix, $v_1, \ldots, v_d$ form an orthogonal basis. Finally, the principal component decomposition of P is $X_P W$, where $W \in \mathbb{R}^{d \times d}$ is the matrix whose columns are $c_1, \ldots, c_d$, in order. For further information on PCA see e.g. [161].

Intuitively, v_i is the variance of the data set in the direction of c_i and consequently, P carries most information in the direction of c_1 and least information in the direction of c_d. By looking at $v_1, \ldots, v_d$, we may notice that there is an $i \in [d-1]$ such that $v_i \gg v_{i+1}$. In this case, we do not lose much information if we (orthogonally) project P onto the first i principal components, which can be carried out by a matrix multiplication $X_P W_i$, where W_i is the matrix consisting only of the first i columns of W. If even $v_{i+1} = 0$, we can achieve an isometry (see Definition 2.2.2) while reducing data. However, since $v_1 \geq \cdots \geq v_d$, the case that $v_1 = \cdots = v_d$ may also occur. (In practice we rather have $v_1 \approx \cdots \approx v_d$.) This means that in contrast to the first example, our data set is inherently high-dimensional and we may not reduce its dimension while maintaining a reasonable error, by using PCA. However, such a case is very unlikely in most applications and PCA is very popular in practice. For example, PCA has been applied to face recognition [183], fingerprint authentication [257], novelty detection (one-class classification) [144], anomaly detection [223], process monitoring [180], medical imaging [141] and signal classification [239] and many more.

In contrast to PCA, (metric) embeddings (see Definition 2.2.3) are not data-dependent, i.e., they can be applied to any point set independently of its intrinsic dimension and always yield a certain guarantee on the distortion of the distances. They have been studied quiet extensively in mathematics in the last centuries, where certain Banach spaces (complete[1] normed spaces) were largely in focus. The most popular examples are the ℓ_p^d **spaces**, which are $\mathbb{R}^d$ equipped with an ℓ_p**-norm**, i.e., $\|(q_1, \dots, q_d)\|_p = \sqrt[p]{\sum_{i=1}^d q_i^p}$, for any $p \in [1, \infty)$. Of course, these spaces are finite-dimensional, but there are also infinite-dimensional variants, which are denoted ℓ_p. These are the sequences $q = (q_1, q_2, \dots) \in \mathbb{R}^{\mathbb{N}}$, equipped with the suitable ℓ_p-norm $\|q\|_p = \sqrt[p]{\sum_{i=1}^\infty q_i^p}$, that satisfy $\|q\|_p < \infty$. If we let $p \to \infty$, we obtain the maximum norm $\|(q_1, \dots, q_d)\|_\infty = \max_{i \in [d]} |q_i|$, respectively $\|(q_1, q_2, \dots)\|_\infty = \max_{i \in \mathbb{N}} |q_i|$, with the corresponding space ℓ_∞^d, respectively ℓ_∞ (see e.g. [226, 195] for further reading).

Let us first note that ℓ_2^d can be considered the Euclidean space, since $\langle p, q \rangle = \frac{1}{4}(\|p+q\|_2^2 - \|p-q\|_2^2)$ (see Section 2.2). Therefore, ℓ_2^d (and also ℓ_2) is also a Hilbert space (a complete inner product space) and in fact, it is the only ℓ_p^d (respectively ℓ_p) space that is. Some further interesting results (see e.g. [195] for further details) in this area are that any finite metric space (X, ρ) can be isometrically embedded into $\ell_\infty^{|X|}$, with (constant) distortion $2q - 1 \geq 3$ into $\ell_\infty^{O(|X|^{1/q} \log |X|)}$, and with distortion $O(\log |X|)$ into ℓ_p^d, for any $p \in [1, \infty)$ and some suitable d. On the negative side, for all $n \in \mathbb{N}_{\geq 2}$ there exists a metric space (X, ρ) with $|X| = n$ that can not be embedded into ℓ_2 with distortion less than $c \frac{\log n}{\log \log n}$, where $c > 0$ is a suitable constant. Furthermore, there exists some (large) value $n \in \mathbb{N}$ and a metric space (X, ρ) with $|X| = n$ that can not be embedded into ℓ_2 with distortion less than $c \log n$, where c is a suitable constant. Note that the aforementioned metric spaces (X, ρ) can be isometrically embedded into $\ell_\infty^{|X|}$, therefore these results also apply to this space. Finally, for each $m \in \mathbb{N}$ with $m \geq 2$ there exists a subset $P \subset \ell_1^m$ with $|P| = 2^m$ (specifically $P = \{0,1\}^m$) that can not be embedded into ℓ_2 with distortion less than $\sqrt{m}$ and furthermore, Bartal et al. [31] have recently proven that for every $p > 2$ there exists an $n \in \mathbb{N}$ and a set $P \subset \ell_p$ with $|P| \geq n$, such that P can not be embedded into ℓ_p^d with distortion less than $(\frac{c \log n}{d})^{\frac{1}{2} - \frac{1}{p}}$, where $c > 0$ is a suitable constant.

We will now review one further and outstanding result on embeddings that we did not yet mention. This result states that any finite subset $P \subset \ell_2^d$ can be embedded *nearly* isometrically into $\ell_2^{d'}$, for some well-chosen d'. Clearly, this result is particularly interesting in computer science, since it allows us to massively reduce data whenever d is much larger than d'. We will see that this is of relevance whenever $d \gg \log |P|$.

5.1.1 The Johnson-Lindenstrauss Embedding

In their 1984 seminal paper [160] on extensions of Lipschitz mappings into a Hilbert space, Johnson and Lindenstrauss introduced a technical lemma, the so-called **Johnson-Lindenstrauss lemma**, which became very popular. This lemma states that for all $\varepsilon \in (0, 1)$ and for any finite set $P \subset \mathbb{R}^d$ there is a function f mapping P into a d'-dimensional linear subspace of $\mathbb{R}^d$, where $d' \in O(\varepsilon^{-2} \log |P|)$, that is an $\frac{1+\varepsilon}{1-\varepsilon}$-embedding with high probability (and an isometry in

[1] A metric space is complete, if every Cauchy sequence of points in the space has a limit also contained in the space. Since we only consider the Euclidean space in this section, which is complete, we omit a formal definition.

expectation). Indeed, a reason that the number of ambient dimensions is often assumed to be constant in analyses of geometric algorithms is because the Johnson-Lindenstrauss lemma can generally be applied, usually yielding a $(1+\varepsilon)$-approximation to the problem while nearly maintaining the asymptotic running time of the method in many cases. We note that in bad cases, which are unfortunately also frequent, a polynomial running time blow-up occurs.

In their proof, Johnson and Lindenstrauss set $f(p) = U^T Q U p$, where $Q \in \mathbb{R}^{d \times d}$ is the matrix (orthogonally) projecting a vector onto its first d' coordinates and U is a matrix chosen uniformly at random from the orthogonal group $\mathcal{O}(d)$ (see Definition 2.2.14). It can easily be verified that f is an orthogonal projection onto a uniformly random d'-dimensional linear subspace (see Definitions 2.2.11 and 2.2.12). However, a drawback of the projection property is that the points still consist of d coordinates each and every coordinate may be necessary, i.e., a dimension can only be ignored if the corresponding coordinate is zero for all points. To circumvent this, modern embeddings in the sense of the Johnson-Lindenstrauss lemma set $f(p) = Up$, where U is a real $d' \times d$ matrix that is drawn from some suitable random distribution.

Here we present a modern variant of the Johnson-Lindenstrauss lemma that uses a matrix whose entries are independent random variables that follow the standard normal distribution. The function does not yield an orthogonal projection (in the strict sense it does not even yield a projection) but approximately preserves distances (and even inner products, cf. Definition 2.2.7 and Theorem 2.2.8) in a similar manner.

Before we do so, we note that there indeed exists a set $P \subset \mathbb{R}^d$ of points, such that $d' \in \Omega(\varepsilon^{-2} \log|P|)$ target dimensions are necessary to maintain a distortion of at most $1+\varepsilon$ [20, 176, 177]. In this sense, the Johnson-Lindenstrauss lemma is (asymptotically) optimal.

Theorem 5.1.1 [41] *For any $\delta, \varepsilon \in (0,1)$ and any (fixed) finite set $P \subset \mathbb{R}^d$ it holds for all $p, q \in P$ that $(1-\varepsilon)\|p-q\| \leq \|f(p)-f(q)\| \leq (1+\varepsilon)\|p-q\|$ with probability at least $1-\delta$ when $f(p) = \frac{Up}{\sqrt{d'}}$ and U is a $d' \times d$ matrix, with $d' \geq 8 \cdot \varepsilon^{-2} \ln\left(\frac{|P|^2}{\delta}\right)$, whose entries are independent standard normal random variables.*

To reveal the underlying technique, we present a succinct variant of the proof in [41].

Proof. Let $V = \{p - q \mid p, q \in P\}$ and $n = |P|$. Since f is linear we can assume that the vectors in V are of unit length. Therefore, in the following we prove (the even stronger statement) that $(1-\varepsilon) \leq \|f(v)\|^2 \leq (1+\varepsilon)$ for all $v \in V$, which implies the claim.

For $v = (v_1, \dots, v_d) \in V$, $i \in [d']$ and $j \in [d]$ we define the random variable $X_{v,i,j} = u_{i,j} \cdot v_j$, where $(u_{i,1}, \dots, u_{i,d})$ is the i^{th} row of U. Since $u_{i,j}$ is standard normally distributed, $X_{v,i,j}$ is normally distributed with zero mean and variance v_j^2 (see Section 2.4.2). We further define $X_{v,i} = \sum_{j=1}^{d} X_{v,i,j}$, which is a normally distributed random variable with zero mean and variance $\sum_{j=1}^{d} v_j^2 = 1$ and thus follows the standard normal distribution. Clearly, $f(v) = 1/\sqrt{d'}(X_{v,1}, \dots, X_{v,d'})$.

We define $X_v = \sum_{i=1}^{d'} X_{v,i}^2$, which is a random variable that follows the Chi-squared distribution with d' degrees of freedom and therefore has mean d' (see Section 2.4.2). By definition, we have $\|f(v)\|^2 = \frac{X_v}{d'}$ and $\operatorname{Exp}[\|f(v)\|^2] = 1 = \|v\|^2$. Furthermore, $|\|f(v)\|^2 - 1| \leq \varepsilon \iff |X_v - d'| \leq \varepsilon d'$.

Using this we can apply Theorem 2.4.18 with $a = \ln\left(\frac{n^2}{\delta}\right)$, which yields for any fixed $v \in V$:

$$\Pr[|X_v - d'| \geq \varepsilon d'] \leq 2 \exp\left(-\ln\left(\frac{n^2}{\delta}\right)\right) = \frac{\delta}{n^2/2},$$

where we use that $d' \geq 8 \cdot \varepsilon^{-2} \ln\left(\frac{n^2}{\delta}\right)$, which implies $2(\sqrt{d'a} + a) \leq \varepsilon d'$.

Finally, by a union bound it holds that $\Pr[\max_{v \in V} |X_v - d'| \geq \varepsilon d'] \leq \delta$, which yields the claim. □

Remark 5.1.2 *While we have presented a slightly weaker variant of the lemma, there are numerous stronger proofs (cf. [80, 113, 4, 182, 164]) of which some rely on other random distributions, like the Rademacher distribution, which are more efficient to sample from, and indeed, $d' \geq 4 \cdot \varepsilon^{-2} \ln(|P|)$ is sufficient such that the error guarantee holds with constant positive probability [80]. Furthermore, any function from one of the various proofs, like the function f, is a $(1+\varepsilon)$-embedding when we rescale ε by $1/3$ (which requires $d' \geq 72 \cdot \varepsilon^{-2} \ln\left(\frac{|P|^2}{\delta}\right)$ in the case of f). For this reason we speak of a* ***Johnson-Lindenstrauss embedding****.*

Johnson-Lindenstrauss embeddings have also extensively been studied empirically and it showed that $d' \geq 2\varepsilon^{-2} \ln(|P|)$ is sufficient in practice [246]. To this day, there is a broad range of applications of these embeddings, e.g. to signal reconstruction [69], regression [121], clustering [42, 167], classification [213], computational linear algebra [229], computational topology [236], graph theory [182], data mining [4], approximate nearest neighbor searching [18] and many more.

While the guarantee of the Johnson-Lindenstrauss embedding immediately extends to sequence distance measures like the dynamic time warping distance and the discrete Fréchet distance by definition, for the continuous Fréchet distance this is not obvious. This is a simple consequence of the fact that distances between points on the line segments that connect the consecutive vertices are also taken into account. Intuitively, a similar guarantee should be possible and indeed, in the following we show that without any knowledge on the employed method, we can derive a similar bound using the guarantee of the Johnson-Lindenstrauss lemma.

We note that one may be tempted to believe that it is sufficient to embed the points corresponding to the *critical values* (see Section 2.7.2) of the curves to achieve an embedding in terms of the Fréchet distance. However, by embedding the curves we practically obtain new curves and their critical values may correspond to points whose preimages must not be points corresponding to the critical values of the original curves. This can be seen from the fact that the employed embedding must not be a projection. Therefore, it is not clear if such an approach yields the desired embedding under the Fréchet distance.

5.1.2 Embedding Polygonal Curves

Now that we have established the fundamentals of embedding points in the Euclidean space via Johnson-Lindenstrauss embeddings, we aim to embed polygonal curves into a lower-dimensional ambient space to improve upon the corresponding running time dependency. We first review some related work.

Related work

The topic of embedding curves (under the Fréchet distance) is relatively unexplored. Only recently Driemel and Krivosija [84] studied the first probabilistic embeddings of the Fréchet distance for c-packed curves, which are curves whose intersections with any ball of radius r are of length at most cr. This class of curves was introduced by Driemel et al. [86] and has so far been considered a viable assumption for realistic curves, see e.g. [12, 51, 83]. In the setting of Driemel and Krivosija, a point p is sampled uniformly at random from the unit sphere in $\mathbb{R}^d$ (centered at the origin), then two curves σ and τ, both of complexity m, are orthogonally projected onto the line through the origin determined by p. They observed that in any case (even if the curves are not c-packed), the discrete Fréchet distance between the curves decreases. Furthermore, they showed that with high probability the discrete Fréchet distance between σ and τ decreases by a factor upper bounded by a function linear in m. Finally, they proved that there exist c-packed curves such that the discrete Fréchet distance decreases by a factor lower bounded by a function linear in m. The latter also holds for the continuous Fréchet distance and for the 1-dynamic time warping distance. They achieve the upper bound by defining and analyzing *guarding sets*, these are subsets of warpings between the curves which determine the discrete Fréchet distance between the projected curves with large probability.

However, the Johnson-Lindenstrauss embedding has been applied to higher-dimensional objects than points before. In a very influential work, Agarwal et al. [11] applied the Johnson-Lindenstrauss embedding to k-dimensional surfaces and curves in $\mathbb{R}^d$ and used these results for the sake of embedding moving points. Similar to Johnson and Lindenstrauss, Agarwal et al. consider orthogonal projections onto a uniformly random d'-dimensional linear subspace. They show that such a projection is a $(1+\varepsilon)$-embedding for any k-dimensional surface of *linearization dimension* δ with positive constant probability, when $d' \in O(\delta\varepsilon^{-2}\log\delta/\varepsilon)$. The linearization dimension roughly corresponds to the (smallest) dimension of an affine subspace that contains the surface. Furthermore, they prove that this can also be achieved for the union of n surfaces of (combined) linearization dimension δ, when $d' \in O(\delta\varepsilon^{-2}\log(n\delta/\varepsilon))$. Weaker results hold for surfaces of bounded *doubling dimension*, i.e., the base two logarithm of the smallest number D, such that a ball of arbitrary radius $r > 0$ can be covered by D balls of radius $r/2$. Here, the pairwise distances of points on the surface can be preserved up to a multiplicative error of $(1 \pm \varepsilon)$ plus an additional additive error of $\pm\varepsilon\Delta$, where Δ is the geodesic diameter of the surface, when $d' \in O(\delta\varepsilon^{-2}\log 1/\varepsilon)$, where δ is an upper bound on the doubling dimension. Furthermore, an analog holds for any curve of length ℓ. Here, the pairwise distances of points on the curve can be preserved up to a multiplicative error of $(1 \pm \varepsilon)$ plus an additional additive error of $\pm\varepsilon\ell$, when $d' \in O(\varepsilon^{-2}\log 1/\varepsilon)$. Agarwal et al. show that no purely multiplicative error-guarantee can be achieved, even if the curve is polygonal.

Magen [190, 191] shows that applying a (scaled) Johnson-Lindenstrauss embedding not only to a given set $P \subset \mathbb{R}^d$ of points, but to $P \cup W$, where $W \subset \mathbb{R}^d$ is a well-chosen set of points determined by P, approximately preserves the height and angles of all triangles determined by any three points in P. Magen even extends this result and shows that by a clever choice of W, the volume (Lebesque measure) of the convex hull of any $k-1$ points from P is approximately preserved when the target dimension is in $\Theta(\varepsilon^{-2}k\log|P|)$. Furthermore, in this case the distance of any point from P to the affine hull of any $k-1$ other points from P is also approximately preserved. These results are finally applied to projective clustering and approximate nearest neighbor (affine subspace) problems.

Another work that inspired our approach is due to Sheehy [236]. He noticed that a Johnson-Lindenstrauss embedding of a set of points yields an embedding for their entire convex hull with additive error. To be precise, for any finite set $P \subset \mathbb{R}^d$, any point $p \in P$ and any point q from the convex hull of P it holds that $|\|f(p)-f(q)\|^2 - \|p-q\|^2| \leq 4\varepsilon r$, when f is a Johnson-Lindenstrauss embedding of P with parameter ε, where r is the radius of the minimum enclosing ball of P.

However, since curves may be drawn apart from each other arbitrarily, this guarantee is not of great use for us and in the following, we extend Johnson-Lindenstrauss type embeddings to polygonal curves and prove a guarantee that only depends on the properties of the curves, namely their number, their complexity and the length of their edges.

Extending Common Methods for Points

First, we aim to extend any method from a pool of common methods for embedding points to a method of embedding polygonal curves. It will show that this extension only nearly yields an embedding guarantee, therefore, in the following we loosen our terminology and also call a function an embedding that yields a combined multiplicative and additive error guarantee on the distances. We generally call this error the distortion.

Since there is a broad family of Johnson-Lindenstrauss type embeddings and any of these can be utilized, we now give a general definition that captures the core property of these embeddings, namely their (probabilistic) error guarantee.

Definition 5.1.3 *Let $P \subset \mathbb{R}^d$ be a finite set. A function $f\colon P \to \mathbb{R}^{d'}$ is a $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding for P, if it holds that*

$$\forall p, q \in P : (1-\varepsilon)\|p-q\| \leq \|f(p)-f(q)\| \leq (1+\varepsilon)\|p-q\|,$$

with constant probability at least $\rho \in (0,1]$ over the random construction of f.

We extend the mapping f from points to polygonal curves by applying it to the vertices of the curves and re-connecting their images in order, see Definition 2.3.9.

Definition 5.1.4 *Let $\tau \in \mathbb{R}^d_*$ be a polygonal curve, $t_1, \ldots, t_m$ be its instants and $v_1, \ldots, v_m$ be its vertices. Let f be a $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding for $\{v_1, \ldots, v_m\}$. By $F(\tau)$ we define the $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding of τ as follows:*

$$F(\tau)(t) = \begin{cases} \operatorname{lp}\left(\overline{f(v_1)f(v_2)}, \frac{t-t_1}{t_2-t_1}\right), & \text{if } t \in [0, t_2) \\ \vdots & \\ \operatorname{lp}\left(\overline{f(v_{m-1})f(v_m)}, \frac{t-t_{m-1}}{t_m-t_{m-1}}\right), & \text{if } t \in [t_{m-1}, 1] \end{cases}.$$

For a set $T = \{\tau_1, \ldots, \tau_n\} \subset \mathbb{R}^d_*$ of polygonal curves we define $F(T) = \{F(\tau) \mid \tau \in T\}$ and require the function f to be a $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding for the set of ***all*** *vertices of* ***all*** *curves $\tau \in T$.*

In the following, we give an explicit bound on the distortion of the Fréchet distance when this function is applied to some given curves. We first express the distance between two points on two line segments by a combination of the distances among their start- and endpoints, using only the relative positions of the points on the respective segment.

Proposition 5.1.5 *Let $s_1 = \overline{p_1p_2}$ and $s_2 = \overline{q_1q_2}$ be line segments between two points $p_1, p_2 \in \mathbb{R}^d$, respectively $q_1, q_2 \in \mathbb{R}^d$. For any $\lambda_p, \lambda_q \in [0,1]$ and $p = \operatorname{lp}(\overline{p_1p_2}, \lambda_p)$ lying on s_1, as well as $q = \operatorname{lp}(\overline{q_1q_2}, \lambda_q)$ lying on s_2, it holds that*

$$\begin{aligned} \|p-q\|^2 = & -(\lambda_p - \lambda_p^2)\|p_1-p_2\|^2 - (\lambda_q - \lambda_q^2)\|q_1-q_2\|^2 + (1-\lambda_p-\lambda_q+\lambda_p\lambda_q)\|p_1-q_1\|^2 \\ & + (\lambda_q - \lambda_p\lambda_q)\|p_1-q_2\|^2 + (\lambda_p-\lambda_p\lambda_q)\|p_2-q_1\|^2 + \lambda_p\lambda_q\|p_2-q_2\|^2. \end{aligned}$$

Proof. We have:

$$
\begin{aligned}
\|p-q\|^2 &= \|(p_1-\lambda_p(p_1-p_2))-(q_1-\lambda_q(q_1-q_2))\|^2 && \text{(I)}\\
&= \|p_1-\lambda_p(p_1-p_2)\|^2 - 2\langle p_1-\lambda_p(p_1-p_2), q_1-\lambda_q(q_1-q_2)\rangle\\
&\quad + \|q_1-\lambda_q(q_1-q_2)\|^2 && \text{(II)}\\
&= \|p_1\|^2 - 2\lambda_p\langle p_1, p_1-p_2\rangle + \lambda_p^2\|p_1-p_2\|^2 - 2\langle p_1-\lambda_p(p_1-p_2), q_1-\lambda_q(q_1-q_2)\rangle\\
&\quad + \|q_1\|^2 - 2\lambda_q\langle q_1, q_1-q_2\rangle + \lambda_q^2\|q_1-q_2\|^2 && \text{(III)}\\
&= \|p_1\|^2 - 2\lambda_p\|p_1\|^2 + 2\lambda_p\langle p_1,p_2\rangle + \lambda_p^2\|p_1\|^2 - 2\lambda_p^2\langle p_1,p_2\rangle + \lambda_p^2\|p_2\|^2\\
&\quad + \|q_1\|^2 - 2\lambda_q\|q_1\|^2 + 2\lambda_q\langle q_1,q_2\rangle + \lambda_q^2\|q_1\|^2 - 2\lambda_q^2\langle q_1,q_2\rangle + \lambda_q^2\|q_2\|^2\\
&\quad - 2(\langle p_1, q_1-\lambda_q(q_1-q_2)\rangle - \langle \lambda_p(p_1-p_2), q_1-\lambda_q(q_1-q_2)\rangle) && \text{(IV)}\\
&= (1-\lambda_p)^2\|p_1\|^2 + \lambda_p^2\|p_2\|^2 + (1-\lambda_q)^2\|q_1\|^2 + \lambda_q^2\|q_2\|^2 + 2\lambda_p(1-\lambda_p)\langle p_1,p_2\rangle\\
&\quad + 2\lambda_q(1-\lambda_q)\langle q_1,q_2\rangle - 2\langle p_1,q_1\rangle + 2\lambda_q\langle p_1,q_1\rangle - 2\lambda_q\langle p_1,q_2\rangle + 2\lambda_p\langle p_1,q_1\rangle\\
&\quad - 2\lambda_p\lambda_q\langle p_1, q_1-q_2\rangle - 2\lambda_p\langle p_2,q_1\rangle + 2\lambda_p\lambda_q\langle p_2, q_1-q_2\rangle && \text{(V)}\\
&= (1-\lambda_p)^2\|p_1\|^2 + \lambda_p^2\|p_2\|^2 + (1-\lambda_q)^2\|q_1\|^2 + \lambda_q^2\|q_2\|^2\\
&\quad + 2\lambda_p(1-\lambda_p)\langle p_1,p_2\rangle + 2(\lambda_p+\lambda_q-\lambda_p\lambda_q-1)\langle p_1,q_1\rangle\\
&\quad + 2(\lambda_p\lambda_q-\lambda_q)\langle p_1,q_2\rangle + 2(\lambda_p\lambda_q-\lambda_p)\langle p_2,q_1\rangle - 2\lambda_p\lambda_q\langle p_2,q_2\rangle\\
&\quad + 2\lambda_q(1-\lambda_q)\langle q_1,q_2\rangle && \text{(VI)}\\
&= (1-\lambda_p)^2\|p_1\|^2 + \lambda_p^2\|p_2\|^2 + (1-\lambda_q)^2\|q_1\|^2 + \lambda_q^2\|q_2\|^2\\
&\quad + 2\lambda_p(1-\lambda_p)\|p_1\|\|p_2\|\cos\sphericalangle(p_1,p_2)\\
&\quad + 2(\lambda_p+\lambda_q-\lambda_p\lambda_q-1)\|p_1\|\|q_1\|\cos\sphericalangle(p_1,q_1)\\
&\quad + 2(\lambda_p\lambda_q-\lambda_q)\|p_1\|\|q_2\|\cos\sphericalangle(p_1,q_2)\\
&\quad + 2(\lambda_p\lambda_q-\lambda_p)\|p_2\|\|q_1\|\cos\sphericalangle(p_2,q_1)\\
&\quad - 2\lambda_p\lambda_q\|p_2\|\|q_2\|\cos\sphericalangle(p_2,q_2) + 2\lambda_q(1-\lambda_q)\|q_1\|\|q_2\|\cos\sphericalangle(q_1,q_2) && \text{(VII)}\\
&= (1-\lambda_p)^2\|p_1\|^2 + \lambda_p^2\|p_2\|^2 + (1-\lambda_q)^2\|q_1\|^2 + \lambda_q^2\|q_2\|^2\\
&\quad + (\lambda_p-\lambda_p^2)(\|p_1\|^2+\|p_2\|^2-\|p_1-p_2\|^2)\\
&\quad + (\lambda_q-\lambda_q^2)(\|q_1\|^2+\|q_2\|^2-\|q_1-q_2\|^2)\\
&\quad - (1-\lambda_p-\lambda_q+\lambda_p\lambda_q)(\|p_1\|^2+\|q_1\|^2-\|p_1-q_1\|^2)\\
&\quad - (\lambda_q-\lambda_p\lambda_q)(\|p_1\|^2+\|q_2\|^2-\|p_1-q_2\|^2)\\
&\quad - (\lambda_p-\lambda_p\lambda_q)(\|p_2\|^2+\|q_1\|^2-\|p_2-q_1\|^2)\\
&\quad - \lambda_p\lambda_q(\|p_2\|^2+\|q_2\|^2-\|p_2-q_2\|^2) && \text{(VIII)}\\
&= (\underbrace{1-1-2\lambda_p+\lambda_p+\lambda_p+\lambda_p^2-\lambda_p^2+\lambda_q-\lambda_q+\lambda_p\lambda_q-\lambda_p\lambda_q}_{=0})\|p_1\|^2\\
&\quad + (\underbrace{\lambda_p^2-\lambda_p^2+\lambda_p-\lambda_p+\lambda_p\lambda_q-\lambda_p\lambda_q}_{=0})\|p_2\|^2\\
&\quad + (\underbrace{1-1-2\lambda_q+\lambda_q+\lambda_q+\lambda_q^2-\lambda_q^2+\lambda_p-\lambda_p+\lambda_p\lambda_q-\lambda_p\lambda_q}_{=0})\|q_1\|^2\\
&\quad + (\underbrace{\lambda_q^2-\lambda_q^2+\lambda_q-\lambda_q+\lambda_p\lambda_q-\lambda_p\lambda_q}_{=0})\|q_2\|^2\\
&\quad - (\lambda_p-\lambda_p^2)\|p_1-p_2\|^2 - (\lambda_q-\lambda_q^2)\|q_1-q_2\|^2 + (1-\lambda_p-\lambda_q+\lambda_p\lambda_q)\|p_1-q_1\|^2\\
&\quad + (\lambda_q-\lambda_p\lambda_q)\|p_1-q_2\|^2 + (\lambda_p-\lambda_p\lambda_q)\|p_2-q_1\|^2 + \lambda_p\lambda_q\|p_2-q_2\|^2. && \text{(IX)}
\end{aligned}
$$

We obtain Eq. (I) to Eq. (VI) using only properties of the Euclidean norm and the dot product, in Eq. (VII) we use the geometric definition of the dot product (derived from Definition 2.2.22) and finally in Eq. (VIII) we apply the law of cosines. Eq. (IX) follows by algebraic manipulations. □

Using Proposition 5.1.5 we can now provide bounds on the distances between the two points on the segments when the segments are embedded in the sense of Definition 5.1.4.

Lemma 5.1.6 *Let $P = \{p_1, \dots, p_n\} \subset \mathbb{R}^d$ be a set of points and f be a $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding for P. Let $p_1, p_2, q_1, q_2 \in P$. For arbitrary $\lambda_p, \lambda_q \in [0,1]$ and $p = \mathrm{lp}\left(\overline{p_1 p_2}, \lambda_p\right)$, $p' = \mathrm{lp}\left(\overline{f(p_1) f(p_2)}, \lambda_p\right)$, as well as $q = \mathrm{lp}\left(\overline{q_1 q_2}, \lambda_q\right)$, $q' = \mathrm{lp}\left(\overline{f(q_1) f(q_2)}, \lambda_q\right)$ it holds that*

$$(1-\varepsilon)^2\|p-q\|^2 - \varepsilon(\|p_1-p_2\|^2+\|q_1-q_2\|^2) \leq \|p'-q'\|^2 \leq (1+\varepsilon)^2\|p-q\|^2 + \varepsilon(\|p_1-p_2\|^2+\|q_1-q_2\|^2)$$

is satisfied with probability at least $\rho \in (0,1]$ over the random construction of f.

Proof. First note that the construction of f succeeds with probability $\rho \in (0,1]$ by Definition 5.1.3. We condition the remaining proof on this event. From Proposition 5.1.5 we now know that

$$\begin{aligned} \|p-q\|^2 = & -(\lambda_p - \lambda_p^2)\|p_1 - p_2\|^2 - (\lambda_q - \lambda_q^2)\|q_1 - q_2\|^2 + (1 - \lambda_p - \lambda_q + \lambda_p\lambda_q)\|p_1 - q_1\|^2 \\ & + (\lambda_q - \lambda_p\lambda_q)\|p_1 - q_2\|^2 + (\lambda_p - \lambda_p\lambda_q)\|p_2 - q_1\|^2 + \lambda_p\lambda_q\|p_2 - q_2\|^2 \end{aligned}$$

and

$$\begin{aligned} \|p'-q'\|^2 = & -(\lambda_p - \lambda_p^2)\|f(p_1) - f(p_2)\|^2 - (\lambda_q - \lambda_q^2)\|f(q_1) - f(q_2)\|^2 \\ & + (1 - \lambda_p - \lambda_q + \lambda_p\lambda_q)\|f(p_1) - f(q_1)\|^2 + (\lambda_q - \lambda_p\lambda_q)\|f(p_1) - f(q_2)\|^2 \\ & + (\lambda_p - \lambda_p\lambda_q)\|f(p_2) - f(q_1)\|^2 + \lambda_p\lambda_q\|f(p_2) - f(q_2)\|^2. \end{aligned}$$

Because every coefficient is non-negative, it can be observed that this sum is maximized under f when

$$\|f(p_1) - f(p_2)\|^2 = (1-\varepsilon)^2\|p_1 - p_2\|^2,$$

$$\|f(q_1) - f(q_2)\|^2 = (1-\varepsilon)^2\|q_1 - q_2\|^2,$$

$$\|f(p_1) - f(q_1)\|^2 = (1+\varepsilon)^2\|p_1 - q_1\|^2,$$

$$\|f(p_1) - f(q_2)\|^2 = (1+\varepsilon)^2\|p_1 - q_2\|^2,$$

$$\|f(p_2) - f(q_1)\|^2 = (1+\varepsilon)^2\|p_2 - q_1\|^2$$

and

$$\|f(p_2) - f(q_2)\|^2 = (1+\varepsilon)^2\|p_2 - q_2\|^2.$$

Using the facts that $(1+\varepsilon)^2 - (1-\varepsilon)^2 = 4\varepsilon$, $(\lambda_q - \lambda_q^2) \leq \frac{1}{4}$ and $(\lambda_p - \lambda_p^2) \leq \frac{1}{4}$, we get that $\|p'-q'\|^2 \leq (1+\varepsilon)^2\|p-q\|^2 + \varepsilon(\|p_1-p_2\|^2+\|q_1-q_2\|^2)$. The lower bound follows analogously. □

These bounds finally yield our main theorem which states the desired error guarantee for the Fréchet distances among the given polygonal curves. Let us first note that these bounds tend to $\mathrm{d_F}(\tau, \sigma)$ when $\varepsilon \to 0$.

Theorem 5.1.7 *Let $T = \{\tau_1, \dots, \tau_n\} \subset \mathbb{R}^d_*$ be a set of polygonal curves and for $\tau \in T$ let $\alpha(\tau)$ denote the maximum distance between two consecutive vertices of τ. Further, for $\tau, \sigma \in T$ let $\alpha(\tau, \sigma) = \max\{\alpha(\tau), \alpha(\sigma)\}$. Now, let F be a $(1 \pm \varepsilon)$-Johnson-Lindenstrauss embedding for T. With constant probability at least $\rho \in (0, 1]$ it holds for all $\tau, \sigma \in T$ that*

$$\sqrt{(1-\varepsilon)^2 \operatorname{d_F}(\tau,\sigma)^2 - 2\varepsilon\alpha(\tau,\sigma)^2} \leq \operatorname{d_F}(F(\tau), F(\sigma)) \leq \sqrt{(1+\varepsilon)^2 \operatorname{d_F}(\tau,\sigma)^2 + 2\varepsilon\alpha(\tau,\sigma)^2},$$

where the exact value for ρ stems from the technique used for obtaining f.

Proof. First note that the construction of f and thus also F succeeds with probability $\rho \in (0, 1]$ by Definition 5.1.3. We condition the remaining proof on this event.

Let $\tau, \sigma \in T$ be arbitrary polygonal curves and $v^\tau_1, \dots, v^\tau_{|\tau|}$, respectively $v^\sigma_1, \dots, v^\sigma_{|\sigma|}$, be their vertices, as well as $t^\tau_1, \dots, t^\tau_{|\tau|}$, respectively $t^\sigma_1, \dots, t^\sigma_{|\sigma|}$, be their instants.

We know from Proposition 2.3.12 that there exist two sequences $(g_k)^\infty_{k=1}$, $(g'_k)^\infty_{k=1}$ in $\mathcal{H}$, such that

$$\lim_{k\to\infty} \max_{t\in[0,1]} \|\tau(t) - \sigma(g_k(t))\| = \operatorname{d_F}(\tau, \sigma),$$

$$\lim_{k\to\infty} \max_{t\in[0,1]} \|F(\tau)(t) - F(\sigma)(g'_k(t))\| = \operatorname{d_F}(F(\tau), F(\sigma)).$$

We can also assume that $\lim\limits_{k\to\infty} |g_k(t) - g_{k-1}(t)| = 0$, respectively $\lim\limits_{k\to\infty} |g'_k(t) - g'_{k-1}(t)| = 0$, for any $t \in [0, 1]$ (see Section 2.7.2).

Further, for each $g \in \mathcal{H}$ and $t \in [0, 1]$ there exists an $i(t) \in \{1, \dots, |\tau| - 1\}$ and a $j(g, t) \in \{1, \dots, |\sigma| - 1\}$ with $t^\tau_{i(t)} \leq t \leq t^\tau_{i(t)+1}$ and $t^\sigma_{j(g,t)} \leq g(t) \leq t^\sigma_{j(g,t)+1}$, such that the following equations hold:

$$F(\tau)(t) = \operatorname{lp}\left(\overline{f(v^\tau_{i(t)})f(v^\tau_{i(t)+1})}, \frac{t - t^\tau_{i(t)}}{t^\tau_{i(t)+1} - t^\tau_{i(t)}}\right),$$

$$F(\sigma)(g(t)) = \operatorname{lp}\left(\overline{f(v^\sigma_{j(g,t)})f(v^\sigma_{j(g,t)+1})}, \frac{g(t) - t^\sigma_{j(g,t)}}{t^\sigma_{j(g,t)+1} - t^\sigma_{j(g,t)}}\right),$$

$$\tau(t) = \operatorname{lp}\left(\overline{v^\tau_{i(t)}v^\tau_{i(t)+1}}, \frac{t - t^\tau_{i(t)}}{t^\tau_{i(t)+1} - t^\tau_{i(t)}}\right),$$

$$\sigma(g(t)) = \operatorname{lp}\left(\overline{v^\sigma_{j(g,t)}v^\sigma_{j(g,t)+1}}, \frac{g(t) - t^\sigma_{j(g,t)}}{t^\sigma_{j(g,t)+1} - t^\sigma_{j(g,t)}}\right).$$

Now, for $g \in \mathcal{H}$, let $t_g \in \arg\max\limits_{t\in[0,1]} \|\sigma(t) - \tau(g(t))\|$ with $\|v^\tau_{i(t)} - v^\tau_{i(t)+1}\| + \|v^\sigma_{j(g,t)} - v^\sigma_{j(g,t)+1}\|$ maximal and let $t'_g \in \arg\max\limits_{t\in[0,1]} \|F(\tau)(t) - F(\sigma)(g(t))\|$ with $\|v^\tau_{i(t)} - v^\tau_{i(t)+1}\| + \|v^\sigma_{j(g,t)} - v^\sigma_{j(g,t)+1}\|$ maximal. It follows immediately that $\lim\limits_{k\to\infty} |t_{g_k} - t_{g_{k-1}}| = 0$, respectively $\lim\limits_{k\to\infty} |t'_{g'_k} - t'_{g'_{k-1}}| = 0$, by definition. Therefore, we argue that all the following limits exist. We now obtain:

$$
\begin{aligned}
\mathrm{d_F}(F(\tau),F(\sigma))^2 &= \lim_{k\to\infty} \|F(\tau)(t'_{g'_k}) - F(\sigma)(g'_k(t'_{g'_k}))\|^2 && \text{(I)}\\
&\leq \lim_{k\to\infty} \|F(\tau)(t'_{g_k}) - F(\sigma)(g_k(t'_{g_k}))\|^2 && \text{(II)}\\
&\leq \lim_{k\to\infty} \Big[(1+\varepsilon)^2 \|\tau(t'_{g_k}) - \sigma(g_k(t'_{g_k}))\|^2 \\
&\qquad + \varepsilon \left(\|v^\tau_{i(t'_{g_k})} - v^\tau_{i(t'_{g_k})+1}\|^2 + \|v^\sigma_{j(g_k,t'_{g_k})} - v^\sigma_{j(g_k,t'_{g_k})+1}\|^2 \right) \Big] && \text{(III)}\\
&\leq \lim_{k\to\infty} \left[(1+\varepsilon)^2 \|\tau(t_{g_k}) - \sigma(g_k(t_{g_k}))\|^2 + 2\varepsilon\alpha(\tau,\sigma)^2\right] && \text{(IV)}\\
&= (1+\varepsilon)^2 \lim_{k\to\infty} \max_{t\in[0,1]} \|\tau(t) - \sigma(g_k(t))\|^2 + 2\varepsilon\alpha(\tau,\sigma)^2\\
&= (1+\varepsilon)^2\, \mathrm{d_F}(\tau,\sigma)^2 + 2\varepsilon\alpha(\tau,\sigma)^2.
\end{aligned}
$$

Eq. (I) follows by definition of g'_k and $t'_{g'_k}$, Eq. (II) follows from the fact that $(g'_k)_{k=1}^\infty$ converges to the infimum and by definitions of g_k and t'_{g_k}, Eq. (III) follows from an application of Lemma 5.1.6 to each element of the sequence and Eq. (IV) follows from the definitions of g_k and t_{g_k} and the definition of $\alpha(\cdot,\cdot)$. The last equation follows from Definition 2.3.11.

Furthermore, we obtain:

$$
\begin{aligned}
\mathrm{d_F}(F(\tau),F(\sigma))^2 &= \lim_{k\to\infty} \|F(\tau)(t'_{g'_k}) - F(\sigma)(g'_k(t'_{g'_k}))\|^2 && \text{(V)}\\
&\geq \lim_{k\to\infty} \|F(\tau)(t_{g'_k}) - F(\sigma)(g'_k(t_{g'_k}))\|^2 && \text{(VI)}\\
&\geq \lim_{k\to\infty} \Big[(1-\varepsilon)^2 \|\tau(t_{g'_k}) - \sigma(g'_k(t_{g'_k}))\|^2 \\
&\qquad - \varepsilon \left(\|v^\tau_{i(t_{g'_k})} - v^\tau_{i(t_{g'_k})+1}\|^2 + \|v^\sigma_{j(g'_k,t_{g'_k})} - v^\sigma_{j(g'_k,t_{g'_k})+1}\|^2 \right) \Big] && \text{(VII)}\\
&\geq \lim_{k\to\infty} \left[(1-\varepsilon)^2 \|\tau(t_{g'_k}) - \sigma(g'_k(t_{g'_k}))\|^2 - 2\varepsilon\alpha(\tau,\sigma)^2\right] && \text{(VIII)}\\
&\geq \lim_{k\to\infty} \left[(1-\varepsilon)^2 \|\tau(t_{g_k}) - \sigma(g_k(t_{g_k}))\|^2 - 2\varepsilon\alpha(\tau,\sigma)^2\right] && \text{(IX)}\\
&= (1-\varepsilon)^2 \lim_{k\to\infty} \max_{t\in[0,1]} \|\tau(t) - \sigma(g_k(t))\|^2 - 2\varepsilon\alpha(\tau,\sigma)^2\\
&= (1-\varepsilon)^2\, \mathrm{d_F}(\tau,\sigma)^2 - 2\varepsilon\alpha(\tau,\sigma)^2.
\end{aligned}
$$

Here, Eq. (V) follows by the definition of g'_k and $t'_{g'_k}$, Eq. (VI) follows, because each element of the sequence is maximized for $t'_{g'_k}$, Eq. (VII) follows from an application of Lemma 5.1.6 to each element of the sequence, Eq. (VIII) follows from the definition of $\alpha(\cdot,\cdot)$ and Eq. (IX) follows from the fact that $(g_k)_{k=1}^\infty$ converges to the infimum. The second last equation follows from the definitions of g_k and t_{g_k} and the last equation follows from Definition 2.3.11. □

Remark 5.1.8 *In [11, Remark 11] Agarwal et al. argue that their bound on the distortion of the pairwise distances between the points on the given curve ([11, Corollary 10]) can not be strengthened to a* true *embedding guarantee, i.e., a purely multiplicative error. Their bound also has an additional additive error, namely $\pm\varepsilon\ell$, where ℓ is the length of the curve. To give proof, they construct a polygonal curve with vertices $0, p_1, p_1 + p_2, \ldots, \sum_{i=1}^n p_i$, where $P = \{p_1, \ldots, p_n\}$ is an ε-net of the unit sphere in $\mathbb{R}^d$ centered at the origin. This curve can only be* truly *embedded, when the distances between each point from P and the origin are preserved. However, their target* *dimension d' is independent of the cardinality of P, and they argue that since $d' < d$ by the rank plus nullity theorem (see [226]) the projection function f, which is linear, has a nullspace of positive dimension. When $\varepsilon \to 0$ a number of points of the ε-net get arbitrarily close to this nullspace, thereby compromising the purely multiplicative error.*

In our case, the target dimension d' depends on the complexities of the curves, therefore we can not adapt the construction in [11, Remark 11]. At this point, we note that one fact that is not yet incorporated into our line of reasoning is that the Fréchet distance between two polygonal curves is not only determined by distances among vertices, but also by distances between a vertex and a point on an edge (see Section 2.7.2) – which does not help to improve our error bound, unfortunately. Interestingly, we can use this fact to argue that a proper additive error must be possible. Assume that the Fréchet distance between two curves σ and τ is determined by the distance of a vertex v^σ of σ to an edge $e^\tau = \overline{v_1^\tau v_2^\tau}$ of τ. Assume that $\|v^\sigma - v_1^\tau\| = \|v^\sigma - v_2^\tau\|$ and therefore, the nearest point on e_τ to v^σ, i.e., the orthogonal projection of v^σ onto e^τ, is $v_p^\sigma = \mathrm{lp}\,(e^\tau, 1/2)$. Now, if by an application of a Johnson-Lindenstrauss embedding f both distances $\|v^\sigma - v_1^\tau\|$ and $\|v^\sigma - v_2^\tau\|$ expand by a factor of $(1+\varepsilon)$ and the distance $\|v_1^\tau - v_2^\tau\|$ expands by a factor of $(1-\varepsilon)$ (contracts), which is possible (see Fig. 5.1), then the nearest point to $v_f^\sigma = f(v^\sigma)$ on $e_f^\tau = \overline{f(v_1^\tau v_2^\tau)}$ is $v_{fp}^\sigma = \mathrm{lp}\left(e_f^\tau, 1/2\right)$ and we obtain

$$\|v_f^\sigma - v_{fp}^\sigma\|^2 = (1+\varepsilon)^2\|v^\sigma - v_1^\tau\|^2 - (1-\varepsilon)^2\frac{\|v_1^\tau - v_2^\tau\|^2}{4}$$

and

$$(1+\varepsilon)^2\|v^\sigma - v_p^\sigma\|^2 = (1+\varepsilon)^2\|v^\sigma - v_1^\tau\|^2 - (1+\varepsilon)^2\frac{\|v_1^\tau - v_2^\tau\|^2}{4}$$

by the law of cosines. Furthermore,

$$\begin{aligned} &\|v_f^\sigma - v_{fp}^\sigma\|^2 > (1+\varepsilon)^2\|v^\sigma - v_p^\sigma\|^2 \\ \iff &(1+\varepsilon)^2\|v^\sigma - v_1^\tau\|^2 - (1-\varepsilon)^2\frac{\|v_1^\tau - v_2^\tau\|^2}{4} > (1+\varepsilon)^2\|v^\sigma - v_1^\tau\|^2 - (1+\varepsilon)^2\frac{\|v_1^\tau - v_2^\tau\|^2}{4} \\ \iff &(1-\varepsilon)\|v_1^\tau - v_2^\tau\| < (1+\varepsilon)\|v_1^\tau - v_2^\tau\|. \end{aligned}$$

Clearly, the last inequality is satisfied for all $\varepsilon \in (0,1)$ whenever $v_1^\tau \neq v_2^\tau$. Furthermore, if the Fréchet distance between $F(\sigma)$ and $F(\tau)$ is determined by the distance between v_f^σ and v_{pf}^σ, then $\mathrm{d_F}(F(\sigma), F(\tau)) > (1+\varepsilon)\,\mathrm{d_F}(\sigma,\tau)$. Clearly, such a situation may arise. For example when τ is a line segment and σ consists of three vertices v_1^σ, v_2^σ and v_2^σ, where v_2^σ has substantially larger distance to τ than v_1^σ and v_3^σ.

We finally note that a similar combined multiplicative and additive error guarantee has already been studied in the literature, in the context of ε-coresets for k-means clustering [27]. These coresets are called *lightweight* coresets and Bachem et al. showed that they have little to no loss in error compared to strong ε-coresets while massively improving upon the running time of the downstream algorithm.

In the following, we empirically study our developed method.

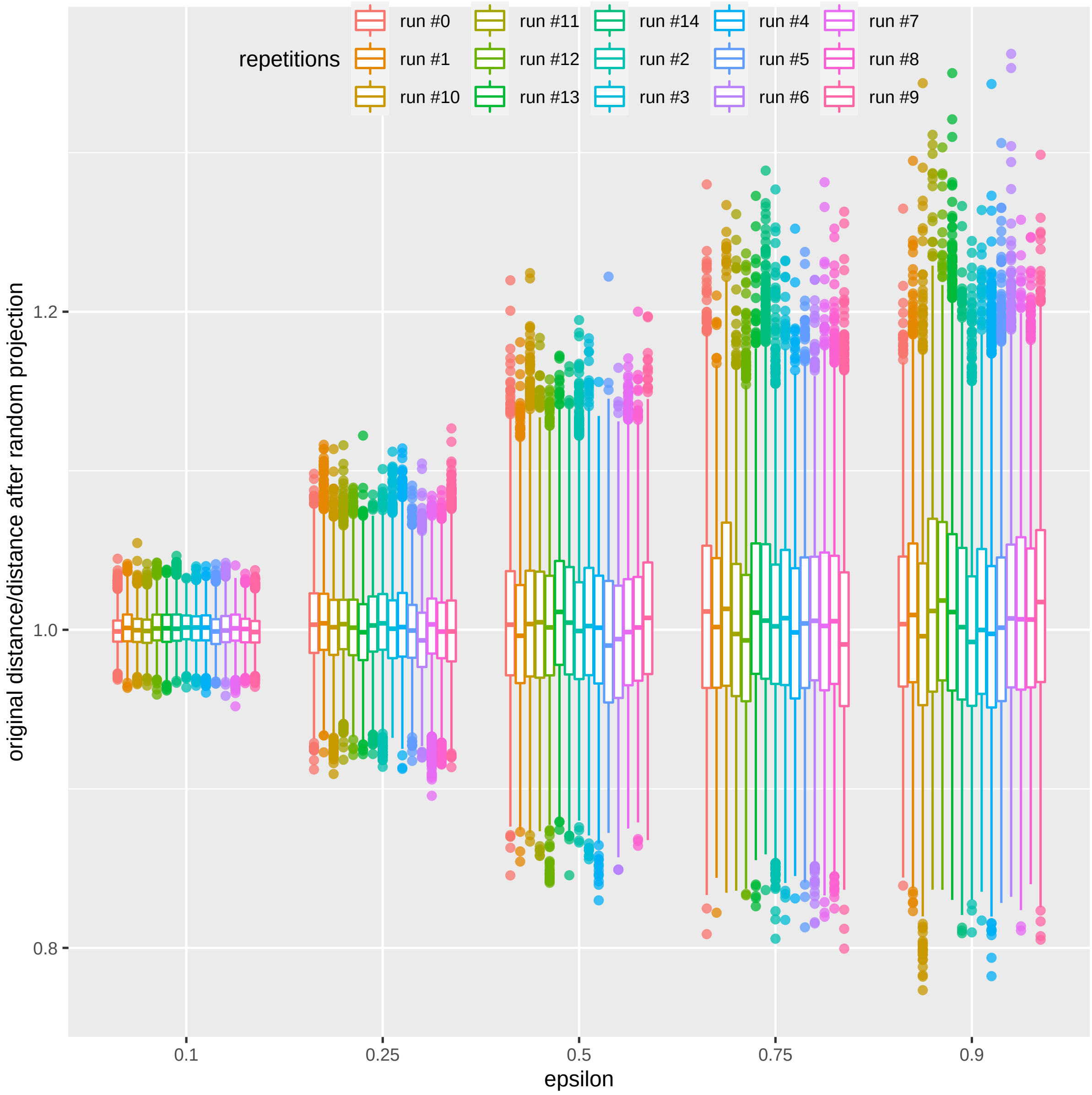

Figure 5.1: For each value of ε we conducted 15 experiments, where we first drew 100 directions uniformly at random, put a point in each of these directions with distance to the origin drawn uniformly at random in the range $[10^{-23}, 10^{3}]$, and then applied a Johnson-Lindenstrauss embedding (based on a matrix with standard normally distributed entries). The plot depicts the change in all pairwise distances, where a value beneath one indicates that the distance contracted and a value above one indicates that the distance expanded. It can be observed that in all experiments, some distances contracted, while others at a time expanded.

Experiments

We experimentally assess the properties of the developed method using a random matrix with independent standard normal distributed entries to embed the vertices of the given curves. This is just the setting that we analyzed in Theorem 5.1.1. We choose the target dimension $d' = \lceil 4 \cdot \varepsilon^{-2} \ln(|P|) \rceil$, according to the proof in [80]. All experiments are conducted using a parallelized C++ implementation of Alt and Godau's algorithm that we developed for this purpose (Fred: fred.dennisrohde.work). We run all experiments on a dedicated virtual machine with 4GB RAM and four (virtual) cores of an Intel Xeon E5-2630 v4 CPU.

We use a data set that comprises cyclical measurements of pressure sensors monitoring the condition of a hydraulic test rig [142]. For a total of six such sensors, it contains 2205 instances (consecutive cycles) of time series where each sensor took measurements at a frequency of 100Hz over the course of 60 seconds. This results in 2205 instances of six time series, each of 6000 values. Such a data set is predestined to appear in applications such as anomaly detection. We choose to build six polygonal curves PS1, ..., PS6 using this data, each of complexity 2205 and 6000 dimensions. In this sense, we interpret each measurement cycle of each sensor as a point in $\mathbb{R}^{6000}$ and by taking the Fréchet distance between two curves we measure the maximum distance between two points from the linear interpolations of the consecutive cycles of the sensors. Intuitively, this is the maximum distance between two cycles measured by two different sensors, among all cycles (interpolated). We think that such a setting may be of interest in practice, e.g. for condition monitoring and especially for anomaly detection.

In 100 reputations each, we compare the CPU time consumed by computing the Fréchet distance between two curves (input $\varepsilon = 0$) and by first embedding the curves and then computing the Fréchet distance (input $\varepsilon > 0$). The results can be seen in Fig. 5.2. It can be observed that for small values of ε, more CPU time is consumed compared to the case that no embedding is done prior to distance computation. This is due to the embedding taking more time to compute than is spared in the distance computation. However, for reasonable values of ε, substantial improvements can be achieved. For example, for $\varepsilon = 0.25$, a speed-up by a factor of 2 is achieved, while for $\varepsilon = 0.9$ even a speed-up by a factor of roughly 16 is achieved.

To demonstrate the effect of parallelization we also repeated the aforementioned experiments and measured the wall time. The results can be seen in Fig. 5.3. It can be observed that the speed-up is proportional to the number of cores.

In both experiments we also measured the relative error achieved by the embedding. The results can be seen in Fig. 5.4. It can be observed that in nearly all cases, the targeted error, or an even better error, could be achieved. Only in three cases, the error guarantee was not met. However, since this guarantee is of *probabilistic* nature, this is in line with the (small) failure probability. We think that the absence of an additive error may be explained by the results of Magen [190, 191]. In detail, he proved that by increasing the target dimension only by a (small) constant factor, all triangles determined by any three points in the given set are approximately preserved. This means that a Johnson-Lindenstrauss embedding preserves with good probability the distances determined by vertex and edge events (see Section 2.7.2).

Finally, we note that it can easily be seen that PCA is not a suitable method for dimension reduction for polygonal curves. To see this, consider the curves σ and τ with vertices $(0,\dots,0)$, $(d,0,\dots,0)$, $(d,d-1,\dots,0)$, $\dots$, $(d,d-1,\dots,1,0)$, respectively $(0,\dots,1)$, $(d,0,\dots,1)$, $(d,d-1,\dots,1)$, $\dots$, $(d,d-1,\dots,1,1)$. Both curves are parallel and their Fréchet distance is one. However, PCA applied to the union of their vertices will identify the d^{th} dimension as least important, while this dimension is the most important for the Fréchet distance.

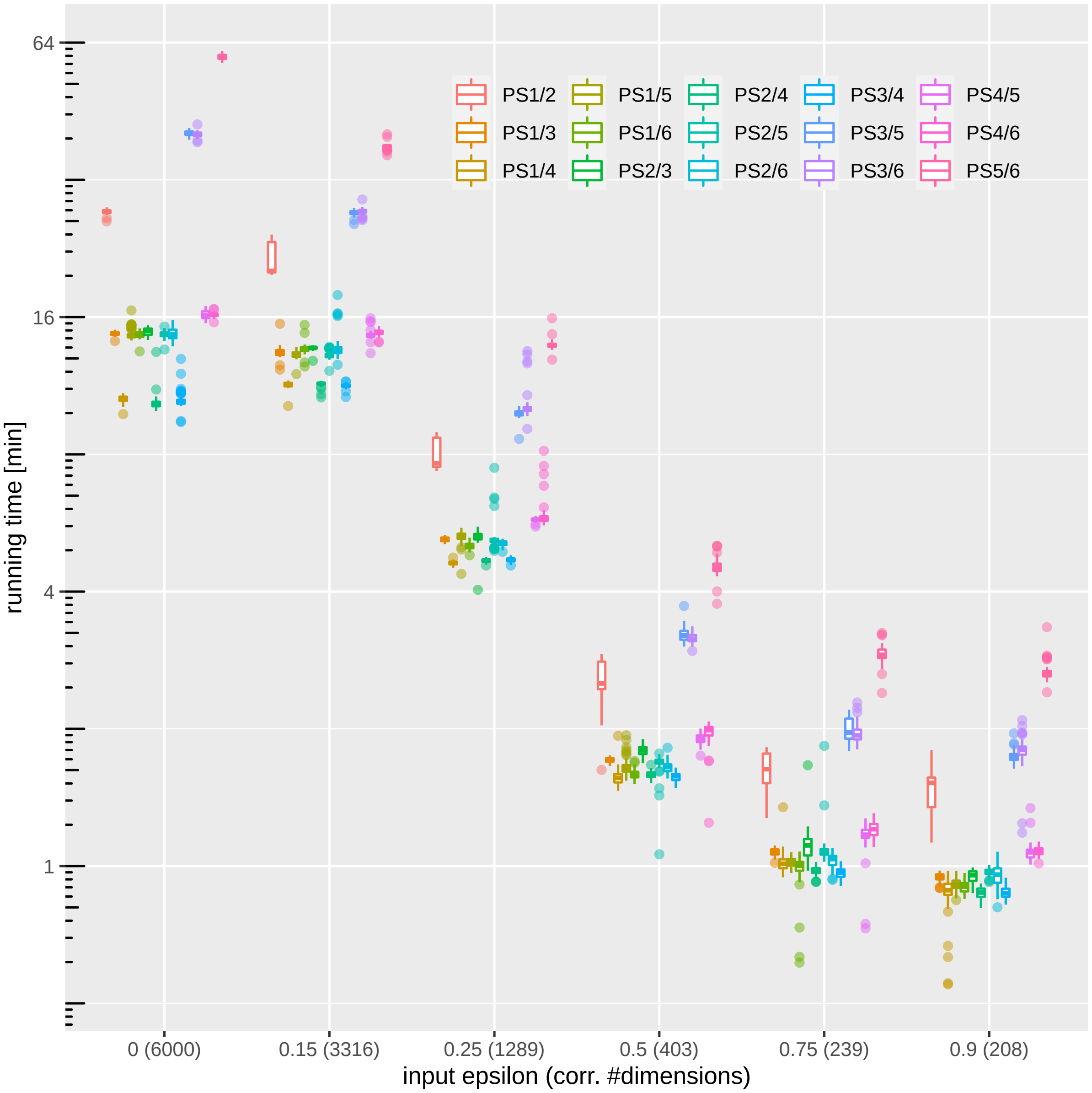

Figure 5.2: In 100 consecutive repetitions each, the CPU time needed to compute the Fréchet distance between two curves, respectively to embed the curves and then compute the Fréchet distance, is measured.

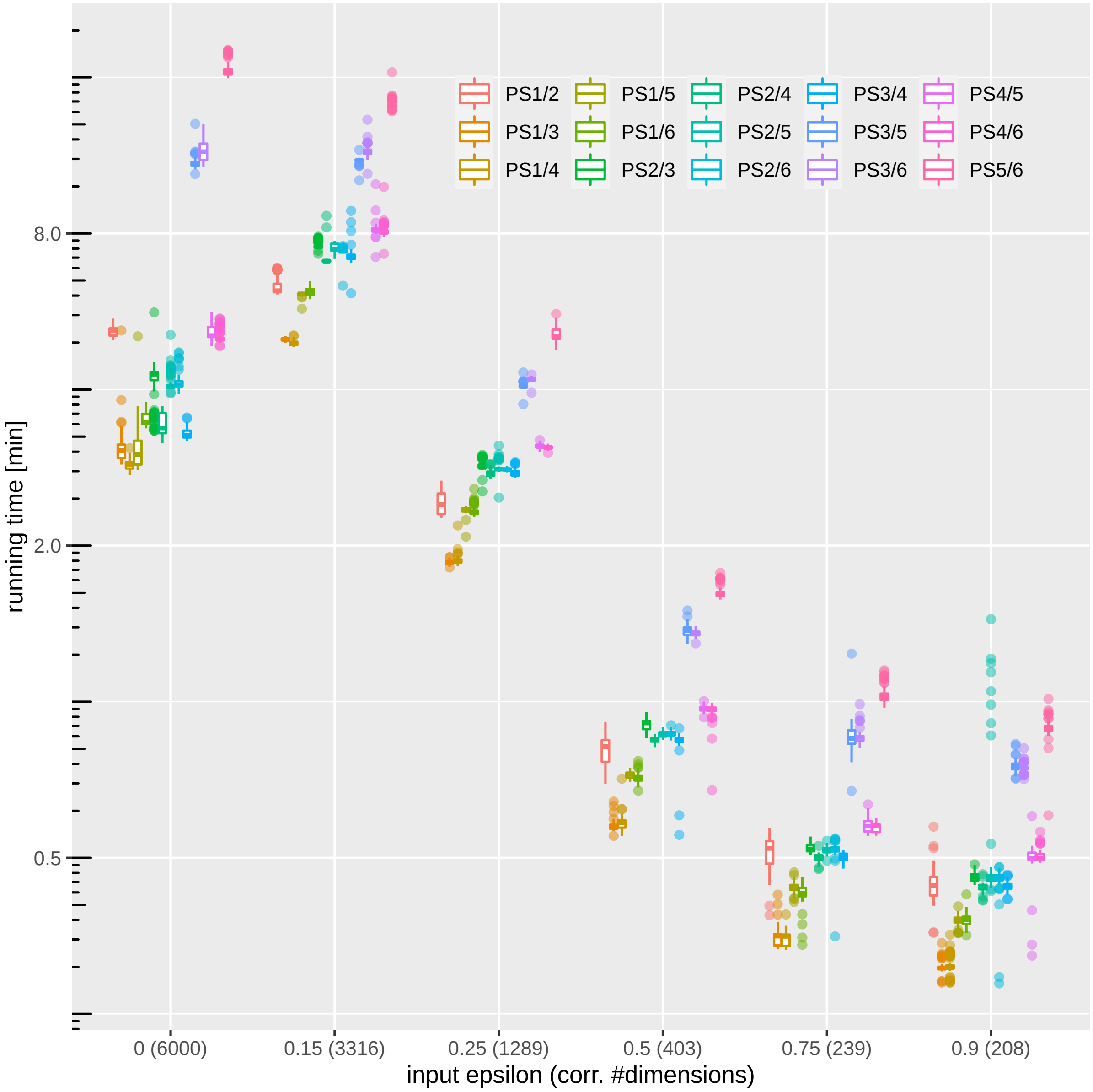

Figure 5.3: In 100 consecutive repetitions each, the (wall) time needed to compute the Fréchet distance between two curves, respectively to embed the curves and then compute the Fréchet distance, is measured.

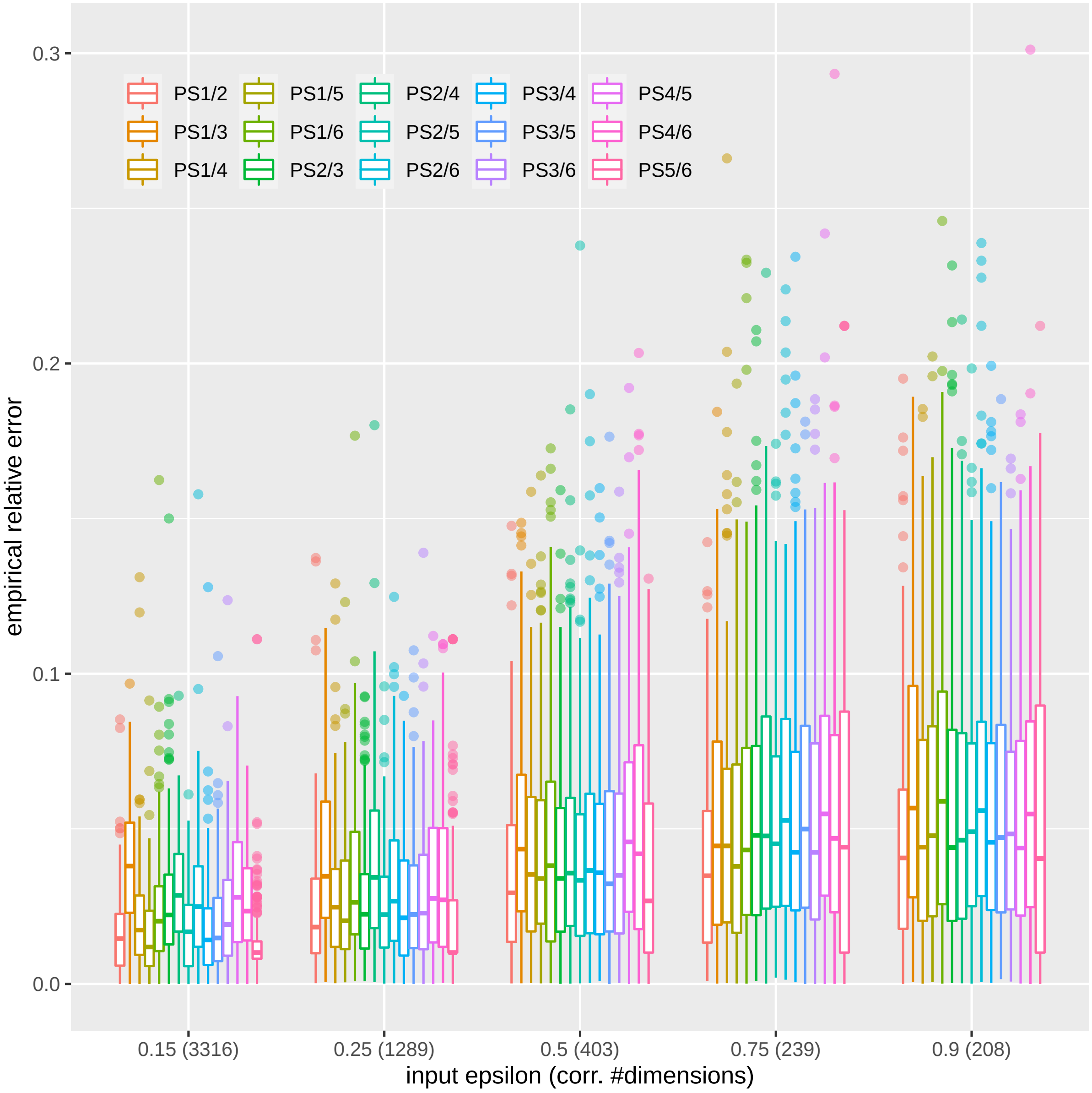

Figure 5.4: In 200 consecutive repetitions each, the distortion of the Fréchet distance between two embedded curves is measured.

5.2 Complexity Reduction for Curves

Recall that Alt and Godau's algorithm is a part of all algorithms considered so far that involve polygonal curves. The running time of this algorithm depends super-quadratic on the complexities of the given polygonal curves. We are interested in improving upon this dependency. An approach to improve upon this dependency is to compress the curves at hand and, if necessary, provide an additional (computationally cheap) function that can be used to (approximately) recover the Fréchet distance between the curves. A way of doing so is by utilizing minimum-complexity r-simplifications. These are another kind of simplifications, and are related to the already considered minimum-error ℓ-simplifications. Here, given a polygonal curve σ, one computes a polygonal curve τ – the simplification of σ – that is of minimum complexity and satisfies $\mathrm{d_F}(\sigma, \tau) \leq r$. For many applications it leads to an efficient constant factor approximation algorithm when we substitute the input curves with their minimum-complexity r-simplifications (with r set to a suitable value). However, as we will see, the complexities of the simplifications can not be bounded in general. The resulting complexity depends on the curve at hand and the given value of r and it can be the case that the complexity does not decrease at all. On a high level, the result of this section is that any reduction of the complexity of a curve may result in the loss of information crucial for determining the Fréchet distance to other curves.

In the following, we generally study the space complexity of compressing a polygonal curve while (approximately) preserving its Fréchet distance to any other curve. We use the tool-chain developed in the field of communication complexity, which was pioneered by Yao [256]. The models studied in this topic comprise at least two parties (the first two are traditionally called *Alice* and *Bob*) and each of them gets a part of the input, denoted by x, respectively y. Their goal is to compute a function $f(x, y)$ of the whole input and to do so, they can send messages to each other and run computations on their part of the input and the messages they have gotten so far. The quantity that is analyzed is the so-called **communication complexity**, i.e., the combined size of the messages (in bits), while the computational running time and space of the involved parties is not central. Typically, lower and upper bounds on the communication complexity of a problem of interest are derived and to obtain lower bounds as general as possible, it is assumed that all parties have unlimited computational power, i.e., they can use infinite running time and space.

The analysis of communication complexity of course requires a formalized procedure that provably leads to the function $f(x, y)$ being computed correctly and which thus determines the messages to share and the computations to carry out. Such a formal procedure is called **communication protocol**, or shortly protocol, and if it only involves computations that can be run in polynomial time, the protocol is *efficient*, otherwise it is *inefficient*. Furthermore, protocols can be either *deterministic*, which means that the involved parties do not have access to randomness, or *randomized*, which means they have. In the latter case, each party may have an own private source of randomness or all parties share a common source of randomness. The source of randomness is an infinite sequence of outcomes from fair coin tosses, from which the parties can read as much as they need. Consequently, in the first case each party has their own sequence of random coin tosses, which only they can see – this setting is the so-called **private coin** model – and in the second case they all can see one public sequence – which is called the **public coin** model. Of course, the second model is more powerful, since parties can see the random coins that the other parties used. However, any protocol in the public coin model can be modified to a protocol in the private coin model while only slightly increasing its complexity, cf. e.g. [174, Theorem 3.14] or [247, Theorem 15.4.6].

In general, if randomness is used, the number of coin tosses read is often also accounted to the communication complexity and protocols do not need to be correct anymore, they only have to succeed with a certain constant positive probability[2]. Also, the number of messages that can be exchanged in a protocol is usually limited. This is formalized by employing communication rounds in which every party can send one message each, and requiring the number of rounds to be bounded by a certain number.

These are the basics of communication complexity and the numerous expressions that go beyond can not be adequately covered by this manuscript. For further reading we therefore refer the reader to [174, 247, 23].

We now formally define the model that we use to study the space complexity of compressing a polygonal curve and approximately recovering its Fréchet distance to any other polygonal curve.

5.2.1 The One-Way Communication Model

A special setting in communication complexity is the so-called one-way communication. The **one-way communication model** (cf. [174, Definition 4.1.7]) consists of two parties, named Alice and Bob, which are given an input x, respectively y, and the protocols are restricted to employ only one round of communication. Specifically, in any protocol Alice first sends an intermediate message to Bob whereupon he sends a solution $s \in f(x, y) \subset \mathbb{R}$ (where f is determined by the problem at hand), which they aimed to recover. This notion of communication complexity is very natural since it connects to fields like statistical learning theory, online computation and streaming algorithms as well as information and coding theory and cryptography, cf. [3, 171, 172, 24, 120, 109].

Since we are here interested in approximately recovering the Fréchet distance between two polygonal curves when one of the curves is compressed, Alice is given a polygonal curve $\sigma \in \mathbb{R}^d_*$ and Bob is given a polygonal curve $\tau \in \mathbb{R}^d_*$, where both are of complexity m. Alice's message consists of a **(one-way) sketch** $S(\sigma)$, which she sends to Bob who then computes an **estimation** $E(S(\sigma), \tau)$, which he sends back to Alice. Their goal is that the estimate approximates the original Fréchet distance between the curves, i.e., $\mathrm{d_F}(\sigma, \tau) \leq E(S(\sigma), \tau) \leq \eta \cdot \mathrm{d_F}(\sigma, \tau)$, where $\eta \in [1, \infty)$ is an approximation factor that parameterizes the problem. In the randomized setting, Alice and Bob may use arbitrarily many public random coins to compute $S(\sigma)$, respectively $E(S(\sigma), \tau)$, and the **one-way communication complexity** of this problem is the minimum number of bits that Alice may use for $S(\sigma)$, such that they succeed (with constant positive probability, if random coins are used) in recovering a solution $s \in [\mathrm{d_F}(\sigma, \tau), \eta \cdot \mathrm{d_F}(\sigma, \tau)]$.

Lower bounds on the communication complexity apply to simplification, sketching protocols and streaming algorithms in general [47]. Upper bounds (by designing protocols) apply to problems like streaming and nearest neighbor search [22]. We now review the work related to the topic.

5.2.2 Related Work

The one-way communication complexity of distance measures has already been studied in the literature, for example of the edit distance [211, 151, 162, 33], Hamming distance [157, 78] and dynamic time warping distance [47]. We now review some selected results and we note that many of the developed protocols are based on (randomized) hashing.

[2] Since one can apply probability boosting, the particular probability is mostly not of interest.

The edit and Hamming distances are measures between two arbitrary finite sequences (from the same domain, $\{0,1\}$ in most cases). The edit distance is the minimum number of insertions, substitutions and deletions to transform the first sequence into the second one and the Hamming distance is the number of positions in which the sequences differ.

The one-way communication complexity of the edit distance has been studied extensively. Here, the so-called *remote file synchronization problem* or shorthand RF_k, has been studied. In this problem a file (represented as a binary string of length n) that was altered on a client is to be updated on a remote server with as little communication as possible. The problem is parameterized by a natural number k and the goal is to reconstruct the altered file when the edit distance to the original is at most k and otherwise to truthfully report that the edit distance is large than k. In particular, solving this problem allows deciding whether the edit distance is at most k or larger. We note that this problem is also often called *document exchange problem* or *correlated files problem*.

It was probably first studied in [211]. Here, an inefficient deterministic one-way protocol is presented that uses a message of $O(k\log(n))$ bits—which is near optimal (cf. [43]). This protocol results from a certain hypergraph coloring approach and Bob's running time is of order $n^{O(k)}$. In [32] an *efficient* deterministic one-way communication protocol with slightly larger message size is presented. This protocol is based on error-correcting codes, particularly Reed-Solomon codes, and Bob's running time is sub-quadratic.

In [151] an efficient randomized one-way protocol is provided that uses a message of almost asymptotically optimal size. This protocol uses an erasure-correcting code, and Bob's running time is sub-quadratic. This result was slightly improved several times in terms of the message size. In [162] a protocol is presented that uses an embedding to the Hamming cube[3] in combination with a sketching procedure for the Hamming distance [218] and a reconstruction procedure. The protocol in [70] is similar. It also uses an embedding to the Hamming cube; using random walks. Currently, the best protocol is due to [33] and builds upon the previously mentioned works.

Concerning, the one-way communication complexity of the Hamming distance, the so-called Gap Hamming distance problem, or shorthand GapHD, has been studied. In this problem one is interested in deciding whether the Hamming distance between two sequences of length n each is at most $n/2-\sqrt{n}$ or at least $n/2+\sqrt{n}$, while it is promised that the distance is not in between $n/2-\sqrt{n}$ and $n/2+\sqrt{n}$. The randomized one-way communication complexity of this problem is in $\Omega(n)$ [254, 157], even though it is already a relaxed variant of deciding the distance.

Finally, concerning the one-way communication complexity of the dynamic time warping distance, the α-DTW and (α,r)-DTW problems have been studied. In both problems one is given two sequences of length at most n and in the α-DTW problem one wants to recover an α-approximation to the DTW distance between the sequences, while the (α,r)-DTW asks to decide whether the DTW distance is at most r or larger than $r\alpha$, for a parameter $r>0$. Both problems are parameterized by an approximation factor $1\leq\alpha\leq n$. Clearly, any protocol for α-DTW can be modified to a protocol for (α,r)-DTW and any lower bound for (α,r)-DTW applies to α-DTW. The main results in [47] are randomized protocols for α-DTW for sequences over a metric space of size polynomial in n and whose ratio between largest and smallest pairwise distance is also bounded by a polynomial in n, and a lower bound for point sequences over a (non-)metric space of at least three elements. The protocols are nearly optimal in the message size and all results are improved for special cases, like sequences over the integers or over separable metric spaces.

We now summarize the aforementioned results.

[3] $\{0,1\}^n$ endowed with the Hamming distance.

Problem	Protocol Type	Efficiency	Bit Complexity	Reference
RF$_k$	deterministic	inefficient	$O(k \log n)$	[211]
		efficient	$O(k^2 + k \log^2 n)$	[32]
	randomized (public)		$O(k \log(n) \log(n/k))$	[151]
			$O(k \log^2(n) \log^*(n))$	[162]
			$O(k^2 \log n)$	[70]
			$O(k \log^2 k + k \log n)$	[33]
GapHD	randomized (public)	$\ast$	$\Omega(n)$	[254]
α-DTW	randomized (public)	inefficient	$O(n/\alpha \log^3 n)$	[47]
		efficient	$O(n/\alpha \log(\alpha) \log^3(n))$	
(α, r)-DTW	randomized (public)	$\ast$	$\Omega(n/\alpha)$	[47]

In the following, we derive lower bounds on the space that is needed to store a representation of a polygonal curve that allows to (approximately) recover the Fréchet distance to any other (uncompressed) polygonal curve.

5.2.3 Lower Bounds by Communication Complexity

First, we prove that the Fréchet distance can not be approximated up to any factor by reducing the complexity of one of the curves at hand deterministically, even in one dimension. We achieve this result by reducing from the equality test communication problem, which requires a linear number of bits. This implies that the Fréchet distance may degrade arbitrarily by (deterministically[4]) performing a proper simplification. Consequently, there is little hope to speed up the Fréchet distance computation, while (approximately) maintaining the original distance, by compressing only one curve at hand.

Theorem 5.2.1 *Let $\sigma, \tau \in \mathbb{R}^d_*$ be polygonal curves with m vertices each. Any deterministic one-way sketch $S(\sigma)$ for which there exists a deterministic estimation $E(S(\sigma), \tau)$ satisfying $\mathrm{d_F}(\tau, \sigma) \leq E(S(\sigma), \tau) \leq \eta \cdot \mathrm{d_F}(\tau, \sigma)$, for $\eta \in [1, \infty)$, consists of $\Omega(m)$ bits.*

Proof. We reduce from the equality test communication problem on bit-strings of size m each. The deterministic communication complexity of this problem is $\Omega(m)$ [247, Theorem 15.2.2].

In this setting Alice and Bob are given bit-strings $A, B \colon [m] \to \{0, 1\}$ and their task is to decide whether there exists at least one $i \in [m]$ such that $A[i] \neq B[i]$ or not with as little communication as possible. We give a one-way protocol for this problem, where only one message from Alice to Bob is allowed.

In a first step, Alice and Bob construct from their bit-strings polygonal curves α, β with $4m$ vertices each. Both curves consist of one gadget per bit. These are either straight-line- or zigzag-gadgets, depending on the value of the respective bit. Specifically, for $i \in [m]$ we define the vertices of α:

If $A[i] = 0$ then $v^\alpha_{4i-3} = 2i$, $v^\alpha_{4i-2} = 2i + 2/3$, $v^\alpha_{4i-1} = 2i + 4/3$ and $v^\alpha_{4i} = 2i + 2$.
Else, if $A[i] = 1$ then $v^\alpha_{4i-3} = 2i$, $v^\alpha_{4i-2} = 2i + 2$, $v^\alpha_{4i-1} = 2i$ and $v^\alpha_{4i} = 2i + 2$.

The vertices $v^\beta_{4i-3}, \ldots, v^\beta_{4i}$ of β are defined analogously.

[4] All known simplification algorithms are deterministic.

We claim that

1. $\exists i \in [m] : A[i] \neq B[i] \implies \mathrm{d_F}(\alpha, \beta) \geq 1$ and
2. $\forall i \in [m] : A[i] = B[i] \implies \mathrm{d_F}(\alpha, \beta) = 0$.

To prove the first item, fix an arbitrary $i \in [m]$. W.l.o.g., assume that $A[i] \neq B[i] = 1$. We have the vertices $v^{\alpha}_{4i-3} = 2i$, $v^{\alpha}_{4i-2} = 2i+2$, $v^{\alpha}_{4i-1} = 2i$ and $v^{\alpha}_{4i} = 2i+2$, as well as, $v^{\beta}_{4i-3} = 2i$, $v^{\beta}_{4i-2} = 2i+2/3$, $v^{\beta}_{4i-1} = 2i+4/3$ and $v^{\beta}_{4i} = 2i+2$. Now, assume that $\mathrm{d_F}(\alpha, \beta) < 1$. This means, that $v^{\alpha}_{4i-3} = 2i$, $v^{\alpha}_{4i-2} = 2i+2$ and $v^{\alpha}_{4i-1} = 2i$ must be mapped to some points that lie closer than $2i+1 \in \overline{v^{\beta}_{4i-2} v^{\beta}_{4i-1}}$. This is a contradiction, because reparameterizations are required to be non-decreasing by definition. Thus, in the optimal case v^{α}_{4i-2} and v^{α}_{4i-1} are mapped to some points infinitesimally close to $2i+1$.

To prove the second item, observe that by symmetry of the construction, α and β represent the same curve and therefore $\mathrm{d_F}(\alpha, \beta) = 0$.

Now, suppose there exist oblivious functions S and E not depending on the data such that $\mathrm{d_F}(\alpha, \beta) \leq E(S(\alpha), \beta) \leq \eta \cdot \mathrm{d_F}(\alpha, \beta)$, for an arbitrary $\eta \in [1, \infty)$.

Alice computes the compressed representation $S(\alpha)$ and communicates $S(\alpha)$ to Bob. Bob evaluates the estimator $E(S(\alpha), \beta)$.

If $E(S(\alpha), \beta) = 0$ then $\mathrm{d_F}(\alpha, \beta) \leq E(S(\alpha), \beta) = 0$.

If $E(S(\alpha), \beta) > 0$ then $\mathrm{d_F}(\alpha, \beta) \geq E(S(\alpha), \beta)/\eta > 0$.

Thus, Bob can distinguish the above two cases and therefore solve the equality test problem, which implies that $S(\alpha)$ consists of $\Omega(m)$ bits. □

Second, we prove that the Fréchet distance can not be approximated within any factor less than $\sqrt{2}$ by reducing the complexity of one of the curves at hand probabilistically. We show this by reducing from the set disjointness communication problem, which also requires a linear number of bits for any randomized protocol succeeding with constant positive probability, cf. [139].

Theorem 5.2.2 *Let $\sigma, \tau \in \mathbb{R}^d_*$ be polygonal curves with m vertices each, where $d \geq 2$. Any randomized one-way sketch $S(\sigma)$ for which there exists a randomized estimation $E(S(\sigma), \tau)$ satisfying $\mathrm{d_F}(\tau, \sigma) \leq E(S(\sigma), \tau) \leq \eta \cdot \mathrm{d_F}(\tau, \sigma)$, for $\eta \in [1, \sqrt{2}]$, consists of $\Omega(m)$ bits.*

Proof. We reduce from the set disjointness communication problem on bit strings of size m each. These represent subsets of a common ground set. The randomized communication complexity with public coins is $\Omega(m)$ [139, Theorem 1.2].

Now, Alice and Bob are given their bit-strings $A, B\colon [m] \to \{0,1\}$ and their task is to decide whether there exists at least one $i \in [m]$ such that $A[i] = B[i] = 1$ or not with as little communication as possible. We give a one-way protocol for this problem, where only one message from Alice to Bob is allowed.

In a first step, Alice and Bob construct from their bit-strings polygonal curves α, β with $4m$ vertices each. Both curves consist of one gadget per bit. These are either straight-line- or notch-gadgets, depending on the value of the respective bit. Thus, for $i \in [m]$ we define the vertices of α:

If $A[i] = 0$ then $v^{\alpha}_{4i-3} = (4i, 0)$, $v^{\alpha}_{4i-2} = (4i, 0)$, $v^{\alpha}_{4i-1} = (4i+4, 0)$ and $v^{\alpha}_{4i} = (4i+4, 0)$. Otherwise, $v^{\alpha}_{4i-3} = (4i, 0)$, $v^{\alpha}_{4i-2} = (4i, 1)$, $v^{\alpha}_{4i-1} = (4i+4, 1)$ and $v^{\alpha}_{4i} = (4i+4, 0)$.

And we define the vertices of β:

If $B[i] = 0$ then $v^{\beta}_{4i-3} = (4i, 0)$, $v^{\beta}_{4i-2} = (4i, 0)$, $v^{\beta}_{4i-1} = (4i+4, 0)$ and $v^{\beta}_{4i} = (4i+4, 0)$. Otherwise, $v^{\beta}_{4i-3} = (4i, 0)$, $v^{\beta}_{4i-2} = (4i, -1)$, $v^{\beta}_{4i-1} = (4i+4, -1)$ and $v^{\beta}_{4i} = (4i+4, 0)$.

We claim that

1. $\exists i \in [m] : (A[i] = B[i] = 1) \implies \mathrm{d_F}(\alpha, \beta) \geq 2$ and
2. $\forall i \in [m] : (A[i] = 0 \vee B[i] = 0) \implies \mathrm{d_F}(\alpha, \beta) < \sqrt{2}$.

To prove the first item, fix an arbitrary $i \in [m]$. If $A[i] = B[i] = 1$, we have the vertices $v^{\alpha}_{4i-3} = (4i, 0)$, $v^{\alpha}_{4i-2} = (4i, 1)$, $v^{\alpha}_{4i-1} = (4i+4, 1)$ and $v^{\alpha}_{4i} = (4i+4, 0)$, as well as, $v^{\beta}_{4i-3} = (4i, 0)$, $v^{\beta}_{4i-2} = (4i, -1)$, $v^{\beta}_{4i-1} = (4i+4, -1)$ and $v^{\beta}_{4i} = (4i+4, 0)$. Now, assume that $\mathrm{d_F}(\alpha, \beta) < 2$. This means, that $(4i+2, 1) \in \overline{v^{\alpha}_{4i-2} v^{\alpha}_{4i-1}}$ must be mapped to some point that lies closer than $(4i+2, -1) \in \overline{v^{\beta}_{4i-2} v^{\beta}_{4i-1}}$. This is a contradiction, because the circle of radius 2 around $(4i+2, 1)$ does only intersect one point of β, namely $(4i+2, -1)$. In particular v^{β}_{4i-3} and v^{β}_{4i} have distance $\sqrt{5} > 2$.

To prove the second item, assume w.l.o.g. that $A[i] \neq B[i]$ for all $i \in [m]$. Otherwise α and β represent the same curve and have distance 0. Let $m = 1$ and w.l.o.g. assume that $B[1] = 1$. Then we have the vertices $v^{\alpha}_{1} = (4, 0)$, $v^{\alpha}_{2} = (4, 0)$, $v^{\alpha}_{3} = (4+4, 0)$ and $v^{\alpha}_{4} = (4+4, 0)$, as well as $v^{\beta}_{1} = (4, 0)$, $v^{\beta}_{2} = (4, -1)$, $v^{\beta}_{3} = (4+4, -1)$ and $v^{\beta}_{4} = (4+4, 0)$. Let g be a reparameterization that maps v^{α}_{1} to v^{β}_{1} and v^{α}_{4} to v^{β}_{4}, as well as $\overline{v^{\beta}_{1} v^{\beta}_{2}}$ and $\overline{v^{\beta}_{3} v^{\beta}_{4}}$ to some infinitesimally small sub-segment of $\overline{v^{\alpha}_{1} v^{\alpha}_{4}}$ each. Since these sub-segments have length less than 1 each, any point of these is mapped to a point within distance less than $\sqrt{2}$. Now, let g map the remaining segment $\overline{v^{\beta}_{2} v^{\beta}_{3}}$ of β linearly to the remaining middle sub-segment of $\overline{v^{\alpha}_{1} v^{\alpha}_{4}}$ of α. Since this remaining sub-segment has length larger than 2, again any point is mapped to a point within distance less than $\sqrt{2}$. Since we can inductively apply this argument for any $m > 1$, i.e., any number of gadgets, we conclude that $\mathrm{d_F}(\alpha, \beta) < \sqrt{2}$.

Now, suppose there exist oblivious randomized functions S and E not depending on the data such that $\mathrm{d_F}(\alpha, \beta) \leq E(S(\alpha), \beta) \leq \eta \cdot \mathrm{d_F}(\alpha, \beta)$ with constant positive probability, for an arbitrary $\eta \in [1, \sqrt{2}]$.

Alice computes the compressed representation $S(\alpha)$ using some public random bits and communicates $S(\alpha)$ to Bob. Bob computes the estimator E, using some public random bits and evaluates $E(S(\alpha), \beta)$.

If $E(S(\alpha), \beta) < 2$ then with constant positive probability $\mathrm{d_F}(\alpha, \beta) \leq E(S(\alpha), \beta) < 2$.

If $E(S(\alpha), \beta) \geq 2$ then with constant positive probability $\mathrm{d_F}(\alpha, \beta) \geq E(S(\alpha), \beta)/\sqrt{2} \geq \sqrt{2}$.

Thus, Bob can distinguish the above two cases and therefore solve the set disjointness problem with constant positive probability, which implies that $S(\alpha)$ consists of $\Omega(m)$ bits. □

We note that it is not known whether this bound is tight. Also, to the best of our knowledge, there are no one-way protocols known for approximating the Fréchet distance, leaving this an open problem.

6 Conclusions

We conclude the thesis by discussing the limits of our results and the problems that remain open.

6.1 Aggregation/Summarization

In Chapter 3 we studied aggregation/summarization problems for polygonal curves and point sequences over some metric space that are derived from the geometric median and (q-)mean problem of points in $\mathbb{R}^d$, namely the ℓ-median and (p,q)-mean/p-mean problems.

For the ℓ-median problem we found an exact algorithm that relies on nondeterminism, but did not find an algorithm that works completely deterministic. This may be explained by the fact that the problem is strongly related to the geometric median problem, for which no exact algorithm exists under standard models of computation. As we turned to approximation algorithms it showed that we needed to rely on grids to obtain small approximation factors. For the resulting running times being independent of the curve's length we introduced shortcutting lemmata, which have the downside that the obtained median curves may have complexity larger than ℓ. We note that the potential approach described in Section 3.1.3 could be used as a post-processing to obtain a median of complexity at most ℓ, though, when the missing components are developed. Anyway, the following problem remains open:

Open Problem 6.1.1 *Does there exist a $(1+\varepsilon)$-approximation algorithm for the ℓ-median problem (in $\mathbb{R}^d$) that returns a polygonal curve of complexity at most ℓ and whose running time depends only on d, ℓ, m, and n?*

Another phenomenon we observed is that by relying on grids the resulting running time became exponential in ℓ, due to a brute-force enumeration. It is not clear to us if, and how, this step may be improved, therefore, the following problem also remains open:

Open Problem 6.1.2 *Does there exist a (bi-criteria) $(1+\varepsilon)$-approximation algorithm for the ℓ-median problem (in $\mathbb{R}^d$) with running time sub-exponential in ℓ?*

For the restricted 2-mean problem on point sequences in the Euclidean space we found a deterministic algorithm solving the problem exactly. We think that this is in line with the geometric mean being determined by an explicit formula. For other values of p, this algorithm can be modified but then yields a $(1+\varepsilon)$-approximation in the best case. Again, this can be explained by algebraic issues – for $p=1$ we again have a strong relation to the geometric median problem. Following, we devised approximation algorithms for the restricted p-mean problem in any metric space and the restricted $(p,1)$-mean problem in the Euclidean space. The running times of these algorithms are all exponential in ℓ, again due to the use of grids. It is not clear whether this dependency is really necessary and the following problem remains open:

Open Problem 6.1.3 *Does there exist an approximation algorithm for any restricted (p,q)-mean problem with running time sub-exponential in ℓ?*

6.2 Clustering

In Chapter 4 we studied the (k, ℓ)-median clustering problem for polygonal curves and the (k, ℓ, p, q)-mean clustering problem for point sequences in an arbitrary metric space. Unfortunately, the running times of our (randomized) approximation algorithms for the (k, ℓ)-median problem are not only exponential in ℓ, but also in k (and also in ε and δ, where ε is a parameter controlling the approximation factor and δ is a parameter controlling the failure probability). Again, this is due to brute-force enumerations – here, of subsets. It is not clear whether these exponential dependencies are necessary, as the (k, ℓ)-median problem is not known to be NP-hard to approximate. The following problem remains open:

Open Problem 6.2.1 *Does there exist an approximation algorithm for the (k, ℓ)-median problem with running time sub-exponential in k?*

As our results for the clustering problems are derived by plugging (modifications of) our results from Chapter 3 into the same generic algorithm, the same problem remains open for the (k, ℓ, p, q)-clustering problem:

Open Problem 6.2.2 *Does there exist an approximation algorithm for the (k, ℓ, p, q)-mean problem with running time sub-exponential in k?*

We also studied the problem of computing ε-coresets for the (k, ℓ)-median problem. Thereby, we proved that sub-linear size ε-coresets exist for this problem and also developed an algorithm that computes these coresets. However, our approach does not work for the (k, ℓ, p, q)-mean problem. Partly, this is due to the non-metric properties of the dynamic time warping distance. In detail, our sensitivity bound relies on the triangle inequality, which does not hold for the dynamic time warping distance, only a weak variant does. It is plausible that a suitable sensitivity bound can be derived using the weak triangle inequality of DTW, but still, to the best of our knowledge the VC dimension of metric balls under the dynamic time warping distance is unknown, which is another missing ingredient. The following problem remains open:

Open Problem 6.2.3 *Do there exist sub-linear size ε-coresets for the (k, ℓ, p, q)-mean problem defined over any metric space?*

Another open problem concerns again the (k, ℓ)-median problem. Here, we could only use our ε-coresets to improve a $(1, \ell)$-median algorithm due to the fact that our (k, ℓ)-median algorithm recursively calls itself on subsets of the input. To improve this algorithm we would need an ε-coreset for any subset of the input, which is clearly prohibitive.

Open Problem 6.2.4 *Can ε-coresets be used to obtain, together with another algorithm, an approximation algorithm for the (k, ℓ)-median problem that improves upon the running times of the known approximation algorithms?*

6.3 Dimension and Complexity Reduction

In Chapter 5 we studied dimension reduction and complexity reduction for polygonal curves that preserve the Fréchet distance. We showed that a straight-forward application of a Johnson-Lindenstrauss embedding to the vertices of polygonal curves *nearly* (up to a possible additive error, which depends on the maximum length of an edge of an involved curve) yields a $(1 + \varepsilon)$-embedding with respect to the Fréchet distance. On real world data, however, this additive error is not present, while it is certainly possible in theory. We think that this may be explained by the work of Magen [190, 191], who showed that a slight increase (by a small constant factor) of the target dimension leads to a preservation of the heights and areas of all triangles determined by three points. In this sense, we think that a Johnson-Lindenstrauss embedding preserves with good probability the vertex and edge events determining the Fréchet distance (see Section 2.7.2). It is not clear though what happens to the monotonicity events. We think that these events do only infrequently appear in real world data, which may explain the absence of the additive error in our experiments. The following problem remains open:

Open Problem 6.3.1 *Is there an upper bound on the target dimension of a Johnson-Lindenstrauss embedding applied on the vertices of a given set of polygonal curves, such that the Fréchet distances among the curves are guaranteed to be preserved up to a multiplicative of* $(1 \pm \varepsilon)$*?*

Finally, by means of communication complexity we have shown that one can not deterministically reduce the complexity of a given polygonal curve and recover a constant factor approximation to the Fréchet distance to any other polygonal curve. For randomized complexity reduction though, we could only rule out $(1 + \varepsilon)$-approximation, for $\varepsilon \in [0, \sqrt{2} - 1]$. In [47], the authors state that Bringmann obtained a sketching protocol for the Fréchet distance achieving approximation. Unfortunately, this work is not published, therefore, the following problem remains open:

Open Problem 6.3.2 *Does there exist an (efficient) protocol for randomized sketching of the Fréchet distance of polygonal curves?*

Bibliography

[1] Waleed H. Abdulla, David Chow, and Gary Sin. Cross-words reference template for DTW-based speech recognition systems. In *TENCON. Conference on Convergent Technologies for Asia-Pacific Region*, volume 4, pages 1576–1579Vol.4, 2003.

[2] Amirali Abdullah, Samira Daruki, and Jeff M. Phillips. Range counting coresets for uncertain data. In Guilherme Dias da Fonseca, Thomas Lewiner, Luis Mariano Peñaranda, Timothy M. Chan, and Rolf Klein, editors, *Symposium on Computational Geometry, SoCG, Rio de Janeiro, Brazil, June 17-20*, pages 223–232. ACM, 2013.

[3] Farid M. Ablayev. Lower Bounds for One-way Probabilistic Communication Complexity. In Andrzej Lingas, Rolf G. Karlsson, and Svante Carlsson, editors, *Automata, Languages and Programming, 20nd International Colloquium, ICALP, Lund, Sweden, July 5-9, Proceedings*, volume 700 of *Lecture Notes in Computer Science*, pages 241–252. Springer, 1993.

[4] Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

[5] Marcel R. Ackermann and Johannes Blömer. Coresets and approximate clustering for Bregman divergences. In Claire Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, New York, NY, USA, January 4-6*, pages 1088–1097. SIAM, 2009.

[6] Marcel R. Ackermann, Johannes Blömer, and Christian Sohler. Clustering for metric and nonmetric distance measures. *ACM Transactions on Algorithms*, 6(4):59:1–59:26, 2010.

[7] Pankaj K. Agarwal and Hai Yu. A space-optimal data-stream algorithm for coresets in the plane. In Jeff Erickson, editor, *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8*, pages 1–10. ACM, 2007.

[8] Pankaj K. Agarwal, Cecilia Magdalena Procopiuc, and Kasturi R. Varadarajan. Approximation Algorithms for k-Line Center. In Rolf H. Möhring and Rajeev Raman, editors, *Algorithms - ESA, 10th Annual European Symposium, Rome, Italy, September 17-21, Proceedings*, volume 2461 of *Lecture Notes in Computer Science*, pages 54–63. Springer, 2002.

[9] Pankaj K. Agarwal, Sariel Har-Peled, and Kasturi R. Varadarajan. Approximating extent measures of points. *Journal of the ACM*, 51(4):606–635, 2004.

[10] Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Robust shape fitting via peeling and grating coresets. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Miami, Florida, USA, January 22-26*, pages 182–191. ACM Press, 2006.

[11] Pankaj K. Agarwal, Sariel Har-Peled, and Hai Yu. Embeddings of Surfaces, Curves, and Moving Points in Euclidean Space. *SIAM Journal on Computing*, 42(2):442–458, 2013.

[12] Pankaj K. Agarwal, Kyle Fox, Jiangwei Pan, and Rex Ying. Approximating Dynamic Time Warping and Edit Distance for a Pair of Point Sequences. In Sándor P. Fekete and Anna Lubiw, editors, *32nd International Symposium on Computational Geometry, SoCG, June 14-18, Boston, MA, USA*, volume 51 of *LIPIcs*, pages 6:1–6:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[13] Charu C. Aggarwal and Chandan K. Reddy, editors. *Data Clustering: Algorithms and Applications.* CRC Press, 2013.

[14] Charu C. Aggarwal and ChengXiang Zhai. A Survey of Text Clustering Algorithms. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 77–128. Springer, 2012.

[15] Hee-Kap Ahn, Helmut Alt, Maike Buchin, Eunjin Oh, Ludmila Scharf, and Carola Wenk. A Middle Curve Based on Discrete Fréchet Distance. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *LATIN: Theoretical Informatics - 12th Latin American Symposium, Ensenada, Mexico, April 11-15, Proceedings*, volume 9644 of *Lecture Notes in Computer Science*, pages 14–26. Springer, 2016.

[16] Hee-Kap Ahn, Helmut Alt, Maike Buchin, Eunjin Oh, Ludmila Scharf, and Carola Wenk. Middle curves based on discrete Fréchet distance. *Computational Geometry*, 89:101621, 2020.

[17] Alfred V. Aho and John E. Hopcroft. *The Design and Analysis of Computer Algorithms.* Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1974.

[18] Nir Ailon and Bernard Chazelle. The Fast Johnson–Lindenstrauss Transform and Approximate Nearest Neighbors. *SIAM Journal on Computing*, 39(1):302–322, 2009.

[19] Daniel Aloise, Amit Deshpande, Pierre Hansen, and Preyas Popat. NP-hardness of Euclidean sum-of-squares clustering. *Machine Learning*, 75(2):245–248, 2009.

[20] Noga Alon. Problems and results in extremal combinatorics–I. *Discrete Mathematics*, 273 (1-3):31–53, 2003.

[21] Helmut Alt and Michael Godau. Computing the Fréchet Distance between two Polygonal Curves. *International Journal of Computational Geometry & Applications*, 5:75–91, 1995.

[22] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David P. Woodruff. Efficient Sketches for Earth-Mover Distance, with Applications. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS, October 25-27, Atlanta, Georgia, USA*, pages 324–330. IEEE Computer Society, 2009.

[23] Sanjeev Arora and Boaz Barak. *Computational Complexity - A Modern Approach.* Cambridge University Press, 2009.

[24] Jan Arpe, Andreas Jakoby, and Maciej Liskiewicz. One-way communication complexity of symmetric Boolean functions. *RAIRO Theoretical Informatics and Applications*, 39(4): 687–706, 2005.

[25] Vijay Arya, Naveen Garg, Rohit Khandekar, Adam Meyerson, Kamesh Munagala, and Vinayaka Pandit. Local Search Heuristics for k-Median and Facility Location Problems. *SIAM Journal on Computing*, 33(3):544–562, 2004.

[26] Olivier Bachem, Mario Lucic, and Andreas Krause. Coresets for Nonparametric Estimation - the Case of DP-Means. In Francis R. Bach and David M. Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning, ICML, Lille, France, 6-11 July*, volume 37 of *JMLR Workshop and Conference Proceedings*, pages 209–217. JMLR.org, 2015.

[27] Olivier Bachem, Mario Lucic, and Andreas Krause. Scalable k-Means Clustering via Lightweight Coresets. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, (KDD)*, pages 1119–1127, 2018.

[28] Chanderjit Bajaj. The algebraic degree of geometric optimization problems. *Discrete & Computational Geometry*, 3(2):177–191, 1988.

[29] Chandrajit L. Bajaj. Proving Geometric Algorithm Non-Solvability: An Application of Factoring Polynomials. *Journal of Symbolic Computation*, 2(1):99–102, 1986.

[30] Sayan Bandyapadhyay, Fedor V. Fomin, and Kirill Simonov. On Coresets for Fair Clustering in Metric and Euclidean Spaces and Their Applications. In Nikhil Bansal, Emanuela Merelli, and James Worrell, editors, *48th International Colloquium on Automata, Languages, and Programming, ICALP, July 12-16, Glasgow, Scotland (Virtual Conference)*, volume 198 of *LIPIcs*, pages 23:1–23:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[31] Yair Bartal, Lee-Ad Gottlieb, and Ofer Neiman. On the Impossibility of Dimension Reduction for Doubling Subsets of ℓ_p. *SIAM Journal on Discrete Mathematics*, 29(3): 1207–1222, 2015.

[32] Djamal Belazzougui. Efficient Deterministic Single Round Document Exchange for Edit Distance. *CoRR*, abs/1511.09229, 2015.

[33] Djamal Belazzougui and Qin Zhang. Edit Distance: Sketching, Streaming, and Document Exchange. In Irit Dinur, editor, *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS, 9-11 October, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 51–60. IEEE Computer Society, 2016.

[34] Jon Louis Bentley and Andrew Chi-Chih Yao. An almost optimal algorithm for unbounded searching. *Information Processing Letters*, 5(3):82–87, 1976.

[35] Marcel Berger, Michael Cole, and Silvio Levy. *Geometry I.* Universitext. Springer Berlin Heidelberg, 2009.

[36] Jiang Bian, Dayong Tian, Yuanyan Tang, and Dacheng Tao. A survey on trajectory clustering analysis. *CoRR*, abs/1802.06971, 2018.

[37] Lenore Blum. *Complexity and real computation.* Springer, 1998.

[38] Lenore Blum, Mike Shub, and Steve Smale. On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions and universal machines. *Bulletin of the American Mathematical Society*, 21:1–46, 1989.

[39] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

[40] Zalán Borsos, Mojmir Mutny, and Andreas Krause. Coresets via Bilevel Optimization for Continual Learning and Streaming. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS, December 6-12, virtual*, 2020.

[41] Stéphane Boucheron, GáborLugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence.* OUP Oxford, 2013.

[42] Christos Boutsidis, Anastasios Zouzias, and Petros Drineas. Random projections for k-means clustering. In John D. Lafferty, Christopher K. I. Williams, John Shawe-Taylor, Richard S. Zemel, and Aron Culotta, editors, *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems. Proceedings of a meeting held 6-9 December, Vancouver, British Columbia Canada*, pages 298–306. Curran Associates, Inc., 2010.

[43] Joshua Brakensiek, Venkatesan Guruswami, and Samuel Zbarsky. Efficient Low-Redundancy Codes for Correcting Multiple Deletions. *IEEE Transactions on Information Theory*, 64 (5):3403–3410, 2018.

[44] Milutin Brankovic, Kevin Buchin, Koen Klaren, André Nusser, Aleksandr Popov, and Sampson Wong. (k, l)-Medians Clustering of Trajectories Using Continuous Dynamic Time Warping. In Chang-Tien Lu, Fusheng Wang, Goce Trajcevski, Yan Huang, Shawn D. Newsam, and Li Xiong, editors, *SIGSPATIAL: 28th International Conference on Advances in Geographic Information Systems, Seattle, WA, USA, November 3-6*, pages 99–110. ACM, 2020.

[45] Vasco Brattka and Peter Hertling. Feasible Real Random Access Machines. *Journal of Complexity*, 14(4):490–526, 1998.

[46] Vladimir Braverman, Dan Feldman, and Harry Lang. New Frameworks for Offline and Streaming Coreset Constructions. *CoRR*, abs/1612.00889, 2016.

[47] Vladimir Braverman, Moses Charikar, William Kuszmaul, David P. Woodruff, and Lin F. Yang. The One-Way Communication Complexity of Dynamic Time Warping Distance. In Gill Barequet and Yusu Wang, editors, *35th International Symposium on Computational Geometry, SoCG, June 18-21, Portland, Oregon, USA*, volume 129 of *LIPIcs*, pages 16:1–16:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[48] Vladimir Braverman, Dan Feldman, Harry Lang, and Daniela Rus. Streaming Coreset Constructions for M-Estimators. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM, September 20-22, Massachusetts Institute of Technology, Cambridge, MA, USA*, volume 145 of *LIPIcs*, pages 62:1–62:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[49] Markus Brill, Till Fluschnik, Vincent Froese, Brijnesh J. Jain, Rolf Niedermeier, and David Schultz. Exact Mean Computation in Dynamic Time Warping Spaces. In Martin Ester and Dino Pedreschi, editors, *Proceedings of the SIAM International Conference on Data Mining, SDM, May 3-5, San Diego Marriott Mission Valley, San Diego, CA, USA*, pages 540–548. SIAM, 2018.

[50] Markus Brill, Till Fluschnik, Vincent Froese, Brijnesh J. Jain, Rolf Niedermeier, and David Schultz. Exact mean computation in dynamic time warping spaces. *Data Mining and Knowledge Discovery*, 33(1):252–291, 2019.

[51] Karl Bringmann. Why Walking the Dog Takes Time: Frechet Distance Has No Strongly Subquadratic Algorithms Unless SETH Fails. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Philadelphia, PA, USA, October 18-21*, pages 661–670. IEEE Computer Society, 2014.

[52] Karl Bringmann and Marvin Künnemann. Quadratic Conditional Lower Bounds for String Problems and Dynamic Time Warping. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS, Berkeley, CA, USA, 17-20 October*, pages 79–97. IEEE Computer Society, 2015.

[53] Hervé Brönnimann, Bernard Chazelle, and Jiri Matousek. Product Range Spaces, Sensitive Sampling, and Derandomization. *SIAM Journal on Computing*, 28(5):1552–1575, 1999.

[54] Kevin Buchin, Maike Buchin, and Yusu Wang. Exact algorithms for partial curve matching via the Fréchet distance. In Claire Mathieu, editor, *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, New York, NY, USA, January 4-6*, pages 645–654. SIAM, 2009.

[55] Kevin Buchin, Maike Buchin, Marc J. van Kreveld, Maarten Löffler, Rodrigo I. Silveira, Carola Wenk, and Lionov Wiratma. Median Trajectories. In Mark de Berg and Ulrich Meyer, editors, *Algorithms - ESA, 18th Annual European Symposium, Liverpool, UK, September 6-8. Proceedings, Part I*, volume 6346 of *Lecture Notes in Computer Science*, pages 463–474. Springer, 2010.

[56] Kevin Buchin, Maike Buchin, Marc J. van Kreveld, Maarten Löffler, Rodrigo I. Silveira, Carola Wenk, and Lionov Wiratma. Median Trajectories. *Algorithmica*, 66(3):595–614, 2013.

[57] Kevin Buchin, Maike Buchin, and Wouter Meulemans andWolfgang Mulzer. Four Soviets Walk the Dog - with an Application to Alt's Conjecture. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Portland, Oregon, USA, January 5-7*, pages 1399–1413. SIAM, 2014.

[58] Kevin Buchin, Maike Buchin, Wouter Meulemans, and Wolfgang Mulzer. Four Soviets Walk the Dog: Improved Bounds for Computing the Fréchet Distance. *Discrete & Computational Geometry*, 58(1):180–216, 2017.

[59] Kevin Buchin, Anne Driemel, Joachim Gudmundsson, Michael Horton, Irina Kostitsyna, Maarten Löffler, and Martijn Struijs. Approximating (k, ℓ)-center clustering for curves. In Timothy M. Chan, editor, *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, San Diego, California, USA, January 6-9*, pages 2922–2938. SIAM, 2019.

[60] Kevin Buchin, Anne Driemel, Natasja van de L'Isle, and André Nusser. klcluster: Center-based Clustering of Trajectories. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 496–499, 2019.

[61] Kevin Buchin, Anne Driemel, and Martijn Struijs. On the Hardness of Computing an Average Curve. In Susanne Albers, editor, *17th Scandinavian Symposium and Workshops on Algorithm Theory*, volume 162 of *LIPIcs*, pages 19:1–19:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[62] Kevin Buchin, André Nusser, and Sampson Wong. Computing Continuous Dynamic Time Warping of Time Series in Polynomial Time. *arXiv e-prints*, art. arXiv:2203.04531, March 2022.

[63] Maike Buchin and Dennis Rohde. Coresets for (k, ℓ)-Median Clustering Under the Fréchet Distance. In Niranjan Balachandran and R. Inkulu, editors, *Algorithms and Discrete Applied Mathematics - 8th International Conference, CALDAM, Puducherry, India, February 10-12, Proceedings*, volume 13179 of *Lecture Notes in Computer Science*, pages 167–180. Springer, 2022.

[64] Maike Buchin, Nicole Funk, and Amer Krivosija. On the complexity of the middle curve problem. *CoRR*, abs/2001.10298, 2020.

[65] Maike Buchin, Anne Driemel, and Dennis Rohde. Approximating (k, ℓ)-Median Clustering for Polygonal Curves. In Dániel Marx, editor, *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, SODA, Virtual Conference, January 10 - 131*, pages 2697–2717. SIAM, 2021.

[66] Maike Buchin, Anne Driemel, Koen van Greevenbroek, Ioannis Psarros, and Dennis Rohde. Approximating Length-Restricted Means under Dynamic Time Warping. *CoRR*, abs/2112.00408, 2021. (to be published).

[67] Peter S. Bullen. *Handbook of Means and Their Inequalities.* Mathematics and Its Applications. Springer Netherlands, 2003.

[68] Laurent Bulteau, Vincent Froese, and Rolf Niedermeier. Tight Hardness Results for Consensus Problems on Circular Strings and Time Series. *SIAM Journal on Discrete Mathematics*, 34(3):1854–1883, 2020.

[69] Emmanuel J Candes and Terence Tao. Near-optimal signal recovery from random projections: Universal encoding strategies? *IEEE Transactions on Information Theory*, 52(12): 5406–5425, 2006.

[70] Diptarka Chakraborty, Elazar Goldenberg, and Michal Koucký. Streaming algorithms for embedding and computing edit distance in the low distance regime. In Daniel Wichs and Yishay Mansour, editors, *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC, Cambridge, MA, USA, June 18-21*, pages 712–725. ACM, 2016.

[71] Erin W. Chambers, Irina Kostitsyna, Maarten Löffler, and Frank Staals. Homotopy Measures for Representative Trajectories. In Piotr Sankowski and Christos D. Zaroliagis, editors, *24th Annual European Symposium on Algorithms, ESA, August 22-24, Aarhus, Denmark*, volume 57 of *LIPIcs*, pages 27:1–27:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[72] Ke 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.

[73] Rachit Chhaya, Anirban Dasgupta, and Supratim Shit. On Coresets for Regularized Regression. In *Proceedings of the 37th International Conference on Machine Learning, ICML, 13-18 July, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1866–1876. PMLR, 2020.

[74] Kenneth L. Clarkson. Coresets, sparse greedy approximation, and the Frank-Wolfe algorithm. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, San Francisco, California, USA, January 20-22*, pages 922–931. SIAM, 2008.

[75] Michael B. Cohen, Yin Tat Lee, Gary Miller, Jakub Pachocki, and Aaron Sidford. Geometric Median in Nearly Linear Time. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, STOC, page 9–21, New York, NY, USA, 2016. Association for Computing Machinery.

[76] George E. Collins. Quantifier elimination for real closed fields by cylindrical algebraic decompostion. In H. Brakhage, editor, *Automata Theory and Formal Languages*, pages 134–183, Berlin, Heidelberg, 1975. Springer Berlin Heidelberg. ISBN 978-3-540-37923-2.

[77] Stephen A. Cook and Robert A. Reckhow. Time-Bounded Random Access Machines. In *Proceedings of the Fourth Annual ACM Symposium on Theory of Computing*, STOC, page 73–80, New York, NY, USA, 1972. Association for Computing Machinery.

[78] Graham Cormode, Mike Paterson, Süleyman Cenk Sahinalp, and Uzi Vishkin. Communication complexity of document exchange. In David B. Shmoys, editor, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, San Francisco, CA, USA*, pages 197–206. ACM/SIAM, 2000.

[79] Anirban Dasgupta, Petros Drineas, Boulos Harb, Ravi Kumar, and Michael W. Mahoney. Sampling algorithms and coresets for ℓ_p regression. In Shang-Hua Teng, editor, *Proceedings of the Nineteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, San Francisco, California, USA, January 20-22*, pages 932–941. SIAM, 2008.

[80] Sanjoy Dasgupta and Anupam Gupta. An elementary proof of a theorem of Johnson and Lindenstrauss. *Random Structures and Algorithms*, 22(1):60–65, January 2003.

[81] Shreyasi Datta, Chandan K. Karmakar, and Marimuthu Palaniswami. Averaging Methods using Dynamic Time Warping for Time Series Classification. In *IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 2794–2798, 2020.

[82] Anne Driemel and Sariel Har-Peled. Jaywalking your dog: computing the Fréchet distance with shortcuts. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Kyoto, Japan, January 17-19*, pages 318–337. SIAM, 2012.

[83] Anne Driemel and Sariel Har-Peled. Jaywalking Your Dog: Computing the Fréchet Distance with Shortcuts. *SIAM Journal on Computing*, 42(5):1830–1866, 2013.

[84] Anne Driemel and Amer Krivosija. Probabilistic Embeddings of the Fréchet Distance. In *Proceedings of the 16th International Workshop on Approximation and Online Algorithms (WAOA)*, pages 218–237, 2018.

[85] Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet distance for realistic curves in near linear time. In David G. Kirkpatrick and Joseph S. B. Mitchell, editors, *Proceedings of the 26th ACM Symposium on Computational Geometry, Snowbird, Utah, USA, June 13-16*, pages 365–374. ACM, 2010.

[86] Anne Driemel, Sariel Har-Peled, and Carola Wenk. Approximating the Fréchet Distance for Realistic Curvesin Near Linear Time. *Discrete & Computational Geometry*, 48(1):94–127, 2012.

[87] Anne Driemel, Amer Krivosija, and Christian Sohler. Clustering time series under the Fréchet distance. In *Proceedings of the Twenty-Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 766–785, 2016.

[88] Anne Driemel, Jeff M. Phillips, and Ioannis Psarros. The VC Dimension of Metric Balls Under Fréchet and Hausdorff Distances. In *35th International Symposium on Computational Geometry*, pages 28:1–28:16, 2019.

[89] Boris A. Dubrovin, Anatoli T. Fomenko, and Sergei P. Novikov. *Modern Geometry — Methods and Applications: Part I: The Geometry of Surfaces, Transformation Groups, and Fields*. Graduate Texts in Mathematics. Springer New York, 1992.

[90] Adrian Dumitrescu and Günter Rote. On the Fréchet distance of a set of curves. In *Proceedings of the 16th Canadian Conference on Computational Geometry, CCCG'04, Concordia University, Montréal, Québec, Canada, August 9-11*, pages 162–165, 2004.

[91] Herbert Edelsbrunner, Joseph O'Rourke, and Raimund Seidel. Constructing Arrangements of Lines and Hyperplanes with Applications. *SIAM Journal on Computing*, 15(2):341–363, May 1986.

[92] Michael Edwards and Kasturi R. Varadarajan. No Coreset, No Cry: II. In Ramaswamy Ramanujam and Sandeep Sen, editors, *FSTTCS: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, Proceedings*, volume 3821 of *Lecture Notes in Computer Science*, pages 107–115. Springer, 2005.

[93] Emre Eftelioglu. Geometric Median. In Shashi Shekhar, Hui Xiong, and Xun Zhou, editors, *Encyclopedia of GIS*, pages 1–4. Springer International Publishing, Cham, 2015.

[94] Peter Egyed and Rephael Wenger. Ordered stabbing of pairwise disjoint convex sets in linear time. *Discrete Applied Mathematics*, 31(2):133–140, 1991.

[95] Thomas Eiter and Heikki Mannila. Computing discrete Fréchet distance. Technical report, Christian Doppler Labor für Expertensyteme, Technische Universität Wien, 1994.

[96] D. Jack Elzinga and Donald W. Hearn. The Minimum Covering Sphere Problem. *Management Science*, 19(1):96–104, 1972.

[97] Jeff Erickson, Ivor van der Hoog, and Tillmann Miltzow. Smoothing the gap between NP and ER. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS, Durham, NC, USA, November 16-19*, pages 1022–1033. IEEE, 2020.

[98] Brian Everitt. *The Cambridge dictionary of statistics.* Cambridge University Press, Cambridge, UK; New York, 2002.

[99] Yawen Fan and Husheng Li. Communication Efficient Coreset Sampling for Distributed Learning. In *19th IEEE International Workshop on Signal Processing Advances in Wireless Communications, SPAWC, Kalamata, Greece, June 25-28*, pages 1–5. IEEE, 2018.

[100] Tomás Feder and Daniel H. Greene. Optimal Algorithms for Approximate Clustering. In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, Chicago, Illinois, USA*, pages 434–444. ACM, 1988.

[101] Dan Feldman and Michael Langberg. A unified framework for approximating and clustering data. In Lance Fortnow and Salil P. Vadhan, editors, *Proceedings of the 43rd ACM Symposium on Theory of Computing*, pages 569–578. ACM, 2011.

[102] Dan Feldman, Amos Fiat, and Micha Sharir. Coresets forWeighted Facilities and Their Applications. In *47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), 21-24 October, Berkeley, California, USA, Proceedings*, pages 315–324. IEEE Computer Society, 2006.

[103] Dan Feldman, Morteza Monemizadeh, and Christian Sohler. A PTAS for k-means clustering based on weak coresets. In Jeff Erickson, editor, *Proceedings of the 23rd ACM Symposium on Computational Geometry, Gyeongju, South Korea, June 6-8*, pages 11–18. ACM, 2007.

[104] Dan Feldman, Morteza Monemizadeh, Christian Sohler, and David P. Woodruff. Coresets and Sketches for High Dimensional Subspace Approximation Problems. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, Austin, Texas, USA, January 17-19*, pages 630–649. SIAM, 2010.

[105] Dan Feldman, Matthew Faulkner, and Andreas Krause. Scalable Training of Mixture Models via Coresets. In John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems. Proceedings of a meeting held 12-14 December, Granada, Spain*, pages 2142–2150, 2011.

[106] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning big data into tiny data: Constant-size coresets for k-means, PCA and projective clustering. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, New Orleans, Louisiana, USA, January 6-8*, pages 1434–1453. SIAM, 2013.

[107] Dan Feldman, Mikhail Volkov, and Daniela Rus. Dimensionality Reduction of Massive Sparse Datasets Using Coresets. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, December 5-10, Barcelona, Spain*, pages 2766–2774, 2016.

[108] Dan Feldman, Melanie Schmidt, and Christian Sohler. Turning Big Data Into Tiny Data: Constant-Size Coresets for k-Means, PCA, and Projective Clustering. *SIAM Journal on Computing*, 49(3):601–657, 2020.

[109] Moran Feldman, Ashkan Norouzi-Fard, Ola Svensson, and Rico Zenklusen. The one-way communication complexity of submodular maximization with applications to streaming and robustness. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC, Chicago, IL, USA, June 22-26*, pages 1363–1374. ACM, 2020.

[110] Hendrik Fichtenberger, Marc Gillé, Melanie Schmidt, Chris Schwiegelshohn, and Christian Sohler. BICO: BIRCH Meets Coresets for k-Means Clustering. In Hans L. Bodlaender and Giuseppe F. Italiano, editors, *Algorithms - ESA - 21st Annual European Symposium, Sophia Antipolis, France, September 2-4. Proceedings*, volume 8125 of *Lecture Notes in Computer Science*, pages 481–492. Springer, 2013.

[111] Gereon Frahling and Christian Sohler. Coresets in dynamic geometric data streams. In Harold N. Gabow and Ronald Fagin, editors, *Proceedings of the 37th Annual ACM Symposium on Theory of Computing, Baltimore, MD, USA, May 22-24*, pages 209–217. ACM, 2005.

[112] Gereon Frahling and Christian Sohler. A fast k-means implementation using coresets. In Nina Amenta and Otfried Cheong, editors, *Proceedings of the 22nd ACM Symposium on Computational Geometry, Sedona, Arizona, USA, June 5-7*, pages 135–143. ACM, 2006.

[113] Péter Frankl and Hiroshi Maehara. The Johnson-Lindenstrauss lemma and the sphericity of some graphs. *Journal of Combinatorial Theory, Series B*, 44(3):355–362, 1988.

[114] M. Maurice Fréchet. Sur quelques points du calcul fonctionnel. *Rendiconti del Circolo Matematico di Palermo (1884-1940)*, 22(1):1–72, December 1906.

[115] Michael L. Fredman and Dan E. Willard. BLASTING through the Information Theoretic Barrier with FUSION TREES. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC, page 1–7, New York, NY, USA, 1990. Association for Computing Machinery.

[116] Michael L. Fredman and Dan E. Willard. Surpassing the Information Theoretic Bound with Fusion Trees. *Journal of Computer and System Sciences*, 47(3):424–436, 1993.

[117] Michael L. Fredman and Dan E. Willard. Trans-dichotomous algorithms for minimum spanning trees and shortest paths. *Journal of Computer and System Sciences*, 48(3): 533–551, 1994.

[118] Guojun Gan, Chaoqun Ma, and Jianhong Wu. *Data Clustering: Theory, Algorithms, and Applications*. Society for Industrial and Applied Mathematics, 2007.

[119] Bernd Gärtner and Martin Jaggi. Coresets for polytope distance. In John Hershberger and Efi Fogel, editors, *Proceedings of the 25th ACM Symposium on Computational Geometry, Aarhus, Denmark, June 8-10*, pages 33–42. ACM, 2009.

[120] Dmitry Gavinsky, Julia Kempe, Iordanis Kerenidis, Ran Raz, and Ronald de Wolf. Exponential separations for one-way quantum communication complexity, with applications to cryptography. In David S. Johnson and Uriel Feige, editors, *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13*, pages 516–525. ACM, 2007.

[121] Leo N. Geppert, Katja Ickstadt, Alexander Munteanu, Jens Quedenfeld, and Christian Sohler. Random projections for Bayesian regression. *Statistics and Computing*, 27(1): 79–101, 2017.

[122] Omer Gold and Micha Sharir. Dynamic Time Warping and Geometric Edit Distance: Breaking the Quadratic Barrier. *ACM Transactions on Algorithms*, 14(4), August 2018.

[123] Teofilo F. Gonzalez. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science*, 38:293–306, 1985.

[124] Pierre Antoine Grillet. *Abstract Algebra*. Graduate Texts in Mathematics. Springer-Verlag, Berlin, Heidelberg, 2007.

[125] Leonidas J. Guibas, John Hershberger, Joseph S. B. Mitchell, and Jack Snoeyink. Approximating Polygons and Subdivisions with Minimum Link Paths. *International Journal of Computational Geometry and Applications*, 3(4):383–415, 1993.

[126] Torben Hagerup. Sorting and Searching on the Word RAM. In *Proceedings of the 15th Annual Symposium on Theoretical Aspects of Computer Science*, STACS, page 366–398, Berlin, Heidelberg, 1998. Springer-Verlag.

[127] Sariel Har-Peled. Clustering Motion. In *42nd Annual Symposium on Foundations of Computer Science, FOCS, 14-17 October, Las Vegas, Nevada, USA*, pages 84–93. IEEE Computer Society, 2001.

[128] Sariel Har-Peled. No, Coreset, No Cry. In Kamal Lodaya and Meena Mahajan, editors, *FSTTCS: Foundations of Software Technology and Theoretical Computer Science, 24th International Conference, Chennai, India, December 16-18, Proceedings*, volume 3328 of *Lecture Notes in Computer Science*, pages 324–335. Springer, 2004.

[129] Sariel Har-Peled. Coresets for Discrete Integration and Clustering. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 33–44. Springer, 2006.

[130] Sariel Har-Peled. *Geometric Approximation Algorithms*. American Mathematical Society, Boston, MA, USA, 2011.

[131] Sariel Har-Peled and Akash Kushal. Smaller coresets for k-median and k-means clustering. In Joseph S. B. Mitchell and Günter Rote, editors, *Proceedings of the 21st ACM Symposium on Computational Geometry, Pisa, Italy, June 6-8*, pages 126–134. ACM, 2005.

[132] Sariel Har-Peled and Akash Kushal. Smaller Coresets for k-Median and k-Means Clustering. *Discrete & Computational Geometry*, 37(1):3–19, 2007.

[133] Sariel Har-Peled and Soham Mazumdar. On coresets for k-means and k-median clustering. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16*, pages 291–300. ACM, 2004.

[134] Sariel Har-Peled and Benjamin Raichel. The frechet distance revisited and extended. In Ferran Hurtado and Marc J. van Kreveld, editors, *Proceedings of the 27th ACM Symposium on Computational Geometry, Paris, France, June 13-15*, pages 448–457. ACM, 2011.

[135] Sariel Har-Peled and Benjamin Raichel. The fréchet distance revisited and extended. *ACM Transactions on Algorithms*, 10(1):3:1–3:22, 2014.

[136] Sariel Har-Peled and Micha Sharir. Relative (p,ε)-Approximations in Geometry. *Discrete & Computational Geometry*, 45(3):462–496, April 2011.

[137] Sariel Har-Peled, Dan Roth, and Dav Zimak. Maximum Margin Coresets for Active and Noise Tolerant Learning. In Manuela M. Veloso, editor, *IJCAI, Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India, January 6-12*, pages 836–841, 2007.

[138] Egbert Harzheim. *Ordered Sets*. Advances in Mathematics. Springer, 2005.

[139] Johan Håstad and Avi Wigderson. The Randomized Communication Complexity of Set Disjointness. *Theory of Computing*, 3(11):211–219, 2007.

[140] Ville Hautamäki, Pekka Nykanen, and Pasi Franti. Time-series clustering by approximate prototypes. In *19th International Conference on Pattern Recognition*, pages 1–4, 2008.

[141] Changtao He, Quanxi Liu, Hongliang Li, and Haixu Wang. Multimodal medical image fusion based on IHS and PCA. *Procedia Engineering*, 7:280–285, 2010. Symposium on Security Detection and Information Processing.

[142] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. Condition monitoring of a complex hydraulic system using multivariate statistics. In *IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*, pages 210–215, May 2015.

[143] John Hershberger and Subhash Suri. Simplified Planar Coresets for Data Streams. In Joachim Gudmundsson, editor, *Algorithm Theory - SWAT, 11th Scandinavian Workshop on Algorithm Theory, Gothenburg, Sweden, July 2-4, Proceedings*, volume 5124 of *Lecture Notes in Computer Science*, pages 5–16. Springer, 2008.

[144] Heiko Hoffmann. Kernel PCA for novelty detection. *Pattern Recognition*, 40(3):863–874, 2007.

[145] Jiawei Huang, Ruomin Huang, Wenjie Liu, Nikolaos M. Freris, and Hu Ding. A Novel Sequential Coreset Method for Gradient Descent Algorithms. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML, 18-24 July, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 4412–4422. PMLR, 2021.

[146] Lingxiao Huang, Shaofeng H.-C. Jiang, Jian Li, and Xuan Wu. Epsilon-Coresets for Clustering (with Outliers) in Doubling Metrics. In *59th IEEE Annual Symposium on Foundations of Computer Science*, pages 814–825. IEEE Computer Society, 2018.

[147] Lingxiao Huang, K. Sudhir, and Nisheeth K. Vishnoi. Coresets for Regressions with Panel Data. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin, editors, *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems, NeurIPS, December 6-12, virtual*, 2020.

[148] Jonathan H. Huggins, Trevor Campbell, and Tamara Broderick. Coresets for Scalable Bayesian Logistic Regression. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems, December 5-10, Barcelona, Spain*, pages 4080–4088, 2016.

[149] Hiroshi Imai and Masao Iri. Polygonal Approximations of a Curve — Formulations and Algorithms. *Machine Intelligence and Pattern Recognition*, 6:71–86, January 1988.

[150] Piotr Indyk. *High-dimensional Computational Geometry.* PhD thesis, Stanford University, CA, USA, 2000.

[151] Utku Irmak, Svilen Mihaylov, and Torsten Suel. Improved single-round protocols for remote file synchronization. In *INFOCOM. 24th Annual Joint Conference of the IEEE Computer and Communications Societies, 13-17 March, Miami, FL, USA*, pages 1665–1676. IEEE, 2005.

[152] Hesam Izakian, Witold Pedrycz, and Iqbal Jamal. Fuzzy clustering of time series data using dynamic time warping distance. *Engineering Applications of Artificial Intelligence*, 39:235–244, 2015.

[153] Anil K. Jain. Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31 (8):651–666, 2010.

[154] Anil K. Jain and Richard C. Dubes. *Algorithms for Clustering Data.* Prentice-Hall, 1988.

[155] Anil K. Jain, M. Narasimha Murty, and Patrick J. Flynn. Data Clustering: A Review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[156] Kamal Jain, Mohammad Mahdian, and Amin Saberi. A new greedy approach for facility location problems. In John H. Reif, editor, *Proceedings on 34^{th} Annual ACM Symposium on Theory of Computing, May 19-21, Montréal, Québec, Canada*, pages 731–740. ACM, 2002.

[157] Thathachar S. Jayram, Ravi Kumar, and Dandapani Sivakumar. The One-Way Communication Complexity of Hamming Distance. *Theory of Computing*, 4(1):129–135, 2008.

[158] Daxin Jiang, Chun Tang, and Aidong Zhang. Cluster Analysis for Gene Expression Data: A Survey. *IEEE Transactions on Knowledge and Data Engineering*, 16(11):1370–1386, 2004.

[159] Norman L. Johnson, Samuel Kotz, and Narayanaswamy Balakrishnan. *Continuous Univariate Distributions.* John Wiley & Sons Incorporated, 2nd edition, 1995.

[160] William B Johnson and Joram Lindenstrauss. Extensions of Lipschitz mappings into a Hilbert space. *Contemporary Mathematics*, 26(1):189–206, 1984.

[161] Ian T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics. Springer, 2002.

[162] Hossein Jowhari. Efficient Communication Protocols for Deciding Edit Distance. In Leah Epstein and Paolo Ferragina, editors, *Algorithms - ESA - 20^{th} Annual European Symposium, Ljubljana, Slovenia, September 10-12. Proceedings*, volume 7501 of *Lecture Notes in Computer Science*, pages 648–658. Springer, 2012.

[163] Praneeth Kacham and David P. Woodruff. Optimal Deterministic Coresets for Ridge Regression. In Silvia Chiappa and Roberto Calandra, editors, *The 23^{rd} International Conference on Artificial Intelligence and Statistics, AISTATS, 26-28 August, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 4141–4150. PMLR, 2020.

[164] Daniel M. Kane and Jelani Nelson. Sparser Johnson-Lindenstrauss Transforms. *Journal of the ACM*, 61(1):4:1–4:23, 2014.

[165] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. A local search approximation algorithm for k-means clustering. *Computational Geometry*, 28(2-3):89–112, 2004.

[166] Zohar S. Karnin and Edo Liberty. Discrepancy, Coresets, and Sketches in Machine Learning. In Alina Beygelzimer and Daniel Hsu, editors, *Conference on Learning Theory, COLT, 25-28 June, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pages 1975–1993. PMLR, 2019.

[167] Michael Kerber and Sharath Raghvendra. Approximation and Streaming Algorithms for Projective Clustering via Random Projections. *CoRR*, abs/1407.2063, 2014.

[168] Lutz Kettner, Kurt Mehlhorn, Sylvain Pion, Stefan Schirra, and Chee-Keng Yap. Classroom examples of robustness problems in geometric computations. *Computational Geometry*, 40 (1):61–78, 2008.

[169] David G. Kirkpatrick and Stefan Reisch. Upper Bounds for Sorting Integers on Random Access Machines. *Theoretical Computer Science*, 28:263–276, 1984.

[170] Matthäus Kleindessner, Pranjal Awasthi, and Jamie Morgenstern. Fair k-Center Clustering for Data Summarization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML, 9-15 June, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 3448–3457. PMLR, 2019.

[171] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. In Frank Thomson Leighton and Allan Borodin, editors, *Proceedings of the Twenty-Seventh Annual ACM Symposium on Theory of Computing, 29 May-1 June, Las Vegas, Nevada, USA*, pages 596–605. ACM, 1995.

[172] Ilan Kremer, Noam Nisan, and Dana Ron. On Randomized One-Round Communication Complexity. *Computational Complexity*, 8(1):21–49, 1999.

[173] Amit Kumar, Yogish Sabharwal, and Sandeep Sen. A Simple Linear Time $(1+\varepsilon)$-Approximation Algorithm for k-Means Clustering in Any Dimensions. In *45th Symposium on Foundations of Computer Science (FOCS), 17-19 October, Rome, Italy, Proceedings*, pages 454–462. IEEE Computer Society, 2004.

[174] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, USA, 1996.

[175] Michael Langberg and Leonard J. Schulman. Universal ε-approximators for Integrals. In *Proceedings of the 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 598–607, 2010.

[176] Kasper Green Larsen and Jelani Nelson. The Johnson-Lindenstrauss Lemma Is Optimal for Linear Dimensionality Reduction. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP, July 11-15, Rome, Italy*, volume 55 of *LIPIcs*, pages 82:1–82:11. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[177] Kasper Green Larsen and Jelani Nelson. Optimality of the Johnson-Lindenstrauss Lemma. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Berkeley, CA, USA, October 15-17*, pages 633–638. IEEE Computer Society, 2017.

[178] Euiwoong Lee, Melanie Schmidt, and John Wright. Improved and simplified inapproximability for k-means. *Information Processing Letters*, 120:40–43, 2017.

[179] Daniel Lemire. Faster retrieval with a two-pass dynamic-time-warping lower bound. *Pattern Recognition*, 42(9):2169–2180, 2009.

[180] Weihua Li, H.Henry Yue, Sergio Valle-Cervantes, and S.Joe Qin. Recursive PCA for adaptive process monitoring. *Journal of Process Control*, 10(5):471–486, 2000.

[181] T. Warren Liao. Clustering of time series data—a survey. *Pattern Recognition*, 38(11): 1857–1874, 2005.

[182] Nathan Linial, Eran London, and Yuri Rabinovich. The geometry of graphs and some of its algorithmic applications. *Combinatorica*, 15(2):215–245, June 1995.

[183] Chengjun Liu. Gabor-Based Kernel PCA with Fractional Power Polynomial Models for Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26 (5):572–581, 2004.

[184] Yu-Tao Liu, Yong'an Zhang, and Ming Zeng. Adaptive Global Time Sequence Averaging Method Using Dynamic Time Warping. *IEEE Transactions on Signal Processing*, 67: 2129–2142, 2019.

[185] Stuart P. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28(2):129–136, 1982.

[186] Hanlin Lu, Ming-Ju Li, Ting He, Shiqiang Wang, Vijaykrishnan Narayanan, and Kevin S. Chan. Robust Coreset Construction for Distributed Machine Learning. In *IEEE Global Communications Conference, GLOBECOM, Waikoloa, HI, USA, December 9-13*, pages 1–6. IEEE, 2019.

[187] Hanlin Lu, Changchang Liu, Shiqiang Wang, Ting He, Vijaykrishnan Narayanan, Kevin S. Chan, and Stephen Pasteris. Joint Coreset Construction and Quantization for Distributed Machine Learning. In *IFIP Networking Conference, Networking, Paris, France, June 22-26*, pages 172–180. IEEE, 2020.

[188] Mario Lucic, Olivier Bachem, and Andreas Krause. Strong Coresets for Hard and Soft Bregman Clustering with Applications to Exponential Family Mixtures. In Arthur Gretton and Christian C. Robert, editors, *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics, AISTATS, Cadiz, Spain, May 9-11*, volume 51 of *JMLR Workshop and Conference Proceedings*, pages 1–9. JMLR.org, 2016.

[189] Jörg Lücke and Dennis Forster. k-means as a variational EM approximation of Gaussian mixture models. *Pattern Recognition Letters*, 125:349–356, 2019.

[190] Avner Magen. Dimensionality Reductions That Preserve Volumes and Distance to Affine Spaces, and Their Algorithmic Applications. In José D. P. Rolim and Salil P. Vadhan, editors, *Randomization and Approximation Techniques, 6th International Workshop, RANDOM, Cambridge, MA, USA, September 13-15, Proceedings*, volume 2483 of *Lecture Notes in Computer Science*, pages 239–253. Springer, 2002.

[191] Avner Magen. Dimensionality Reductions in l_2 that Preserve Volumes and Distance to Affine Spaces. *Discrete & Computational Geometry*, 38(1):139–153, 2007.

[192] Meena Mahajan, Prajakta Nimbhorkar, and Kasturi R. Varadarajan. The planar k-means problem is NP-hard. *Theoretical Computer Science*, 442:13–21, 2012.

[193] Anil Maheshwari, Jörg-Rüdiger Sack, and Christian Scheffer. Approximating the integral Fréchet distance. *Computational Geometry*, 70-71:13–30, 2018.

[194] Harry G. Mairson and Jorge Stolfi. Reporting and Counting Intersections Between Two Sets of Line Segments. In Rae A. Earnshaw, editor, *Theoretical Foundations of Computer Graphics and CAD*, pages 307–325, Berlin, Heidelberg, 1988. Springer.

[195] Jirí Matousek. *Lectures on discrete geometry*, volume 212 of *Graduate texts in mathematics*. Springer, 2002.

[196] Jirí Matousek. Intersection graphs of segments and $\exists\mathbb{R}$. *CoRR*, abs/1406.2636, 2014.

[197] Nimrod Megiddo and Kenneth J. Supowit. On the Complexity of Some Common Geometric Location Problems. *SIAM Journal on Computing*, 13(1):182–196, 1984.

[198] Stefan Meintrup, Alexander Munteanu, and Dennis Rohde. Random Projections and Sampling Algorithms for Clustering of High-Dimensional Polygonal Curves. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems, NeurIPS, December 8-14, Vancouver, BC, Canada*, pages 12807–12817, 2019.

[199] Avraham Melkman and Joseph O'Rourke. On Polygonal Chain Approximation. In Godfried T. Toussaint, editor, *Computational Morphology*, volume 6 of *Machine Intelligence and Pattern Recognition*, pages 87–95. North-Holland, 1988.

[200] Z.A. Melzak. *Companion to Concrete Mathematics: Mathematical Techniques and Various Applications*. Wiley, 1973.

[201] Baharan Mirzasoleiman, Jeff A. Bilmes, and Jure Leskovec. Coresets for Data-efficient Training of Machine Learning Models. In *Proceedings of the 37th International Conference on Machine Learning, ICML, 13-18 July, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 6950–6960. PMLR, 2020.

[202] Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Randomization and Probabilistic Techniques in Algorithms and Data Analysis*. Cambridge University Press, USA, 2nd edition, 2017.

[203] Marion Morel, Catherine Achard, Richard Kulpa, and Séverine Dubuisson. Time-series averaging using constrained dynamic time warping with tolerance. *Pattern Recognition*, 74: 77–89, 2018.

[204] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[205] Alexander Munteanu and Chris Schwiegelshohn. Coresets – Methods and History: A Theoreticians Design Pattern for Approximation and Streaming Algorithms. *Künstliche Intelligenz*, 32(1):37–53, 2018.

[206] Alexander Munteanu, Chris Schwiegelshohn, Christian Sohler, and David P. Woodruff. On Coresets for Logistic Regression. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems, NeurIPS, December 3-8, Montréal, Canada*, pages 6562–6571, 2018.

[207] Abhinandan Nath and Erin Taylor. k-Median Clustering Under Discrete Fréchet and Hausdorff Distances. In Sergio Cabello and Danny Z. Chen, editors, *36th International Symposium on Computational Geometry*, volume 164 of *LIPIcs*, pages 58:1–58:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[208] Vit Niennattrakul and Chotirat Ann Ratanamahatana. Inaccuracies of Shape Averaging Method Using Dynamic Time Warping for Time Series Data. In Yong Shi, G. Dick van Albada, Jack J. Dongarra, and Peter M. A. Sloot, editors, *Computational Science - ICCS, 7th International Conference Beijing, China, May 27-30, Proceedings, Part I*, volume 4487 of *Lecture Notes in Computer Science*, pages 513–520. Springer, 2007.

[209] Vit Niennattrakul and Chotirat Ann Ratanamahatana. Shape averaging under Time Warping. In *6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*, volume 02, pages 626–629, 2009.

[210] Manabu Okawa. Time-series averaging and local stability-weighted dynamic time warping for online signature verification. *Pattern Recognition*, 112:107699, 2021.

[211] Alon Orlitsky. Interactive Communication: Balanced Distributions, Correlated Files, and Average-Case Complexity. In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October*, pages 228–238. IEEE Computer Society, 1991.

[212] Susan Hesse Owen and Mark S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.

[213] Saurabh Paul, Christos Boutsidis, Malik Magdon-Ismail, and Petros Drineas. Random Projections for Linear Support Vector Machines. *ACM Transactions on Knowledge Discovery from Data*, 8(4):22:1–22:25, 2014.

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

[215] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[216] François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn J. Keogh. Dynamic Time Warping Averaging of Time Series Allows Faster and More Accurate Classification. In Ravi Kumar, Hannu Toivonen, Jian Pei, Joshua Zhexue Huang, and Xindong Wu, editors, *IEEE International Conference on Data Mining, ICDM, Shenzhen, China, December 14-17*, pages 470–479. IEEE Computer Society, 2014.

[217] Jeff M. Phillips and Wai Ming Tai. Improved Coresets for Kernel Density Estimates. In Artur Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, New Orleans, LA, USA, January 7-10*, pages 2718–2727. SIAM, 2018.

[218] Ely Porat and Ohad Lipsky. Improved Sketching of Hamming Distance with Error Correcting. In Bin Ma and Kaizhong Zhang, editors, *Combinatorial Pattern Matching, 18th Annual Symposium, CPM, London, Canada, July 9-11, Proceedings*, volume 4580 of *Lecture Notes in Computer Science*, pages 173–182. Springer, 2007.

[219] F.P. Preparata and M.I. Shamos. *Computational Geometry: An Introduction.* Monographs in Computer Science. Springer New York, 1985.

[220] Lawrence Rabiner and Jay Wilpon. Considerations in applying clustering techniques to speaker independent word recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 4, pages 578–581, 1979.

[221] Edgar A. Ramos. Intersection of Unit-balls and Diameter of a Point Set in $\mathbb{R}^3$. *Computational Geometry*, 8:57–65, 1997.

[222] Sashank J. Reddi, Barnabás Póczos, and Alexander J. Smola. Communication Efficient Coresets for Empirical Loss Minimization. In Marina Meila and Tom Heskes, editors, *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence, UAI, July 12-16, Amsterdam, The Netherlands*, pages 752–761. AUAI Press, 2015.

[223] Haakon Ringberg, Augustin Soule, Jennifer Rexford, and Christophe Diot. Sensitivity of PCA for traffic anomaly detection. In Leana Golubchik, Mostafa H. Ammar, and Mor Harchol-Balter, editors, *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS, San Diego, California, USA, June 12-16*, pages 109–120. ACM, 2007.

[224] Nery Riquelme-Granada, Khuong Nguyen, and Zhiyuan Luo. Coreset-based Conformal Prediction for Large-scale Learning. In Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, and Evgueni N. Smirnov, editors, *Conformal and Probabilistic Prediction and Applications, COPA, 9-11 September, Golden Sands, Bulgaria*, volume 105 of *Proceedings of Machine Learning Research*, pages 142–162. PMLR, 2019.

[225] Borut Robic. *The Foundations of Computability Theory.* Springer Publishing Company, Incorporated, 1st edition, 2015.

[226] Steven Roman. *Advanced Linear Algebra*, volume 135 of *Graduate texts in mathematics.* Springer New York, New York, NY, 2008.

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

[228] David Salesin, Jorge Stolfi, and Leonidas J. Guibas. Epsilon Geometry: Building Robust Algorithms from Imprecise Computations. In Kurt Mehlhorn, editor, *Proceedings of the Fifth Annual Symposium on Computational Geometry, Saarbrücken, Germany, June 5-7*, pages 208–217. ACM, 1989.

[229] Tamás Sarlós. Improved Approximation Algorithms for Large Matrices via Random Projections. In *Proceedings of the 47^{th} Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 143–152, 2006.

[230] Nathan Schaar, Vincent Froese, and Rolf Niedermeier. Faster Binary Mean Computation Under Dynamic Time Warping. In Inge Li Gørtz and Oren Weimann, editors, *31st Annual Symposium on Combinatorial Pattern Matching, CPM, June 17-19, Copenhagen, Denmark*, volume 161 of *LIPIcs*, pages 28:1–28:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[231] Marcus Schaefer and Daniel Stefankovic. Fixed Points, Nash Equilibria, and the Existential Theory of the Reals. *Theory of Computing Systems*, 60(2):172–193, 2017.

[232] Arnold Schönhage. On the power of random access machines. In *Automata, Languages and Programming.*, volume 71 of *Lecture Notes in Computer Science*, pages 520–529. Springer-Verlag, 1979.

[233] David Schultz and Brijnesh Jain. Nonsmooth analysis and subgradient methods for averaging in dynamic time warping spaces. *Pattern Recognition*, 74:340–358, 2018.

[234] Mícheál Ó Searcóid. *Metric Spaces.* Springer Undergraduate Mathematics Series. Springer London, 2006.

[235] Michael I. Shamos. *Computational Geometry.* PhD thesis, Yale University, New Haven, CT, USA, 1978.

[236] Donald R. Sheehy. The Persistent Homology of Distance Functions under Random Projection. In *30th Annual Symposium on Computational Geometry, (SoCG)*, page 328, 2014.

[237] Albert N. Shiryaev. *Probability.* Graduate Texts in Mathematics. Springer-Verlag, New York, 2 edition, 1996.

[238] Christian Sohler and David P. Woodruff. Strong Coresets for k-Median and Subspace Approximation: Goodbye Dimension. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS, Paris, France, October 7-9*, pages 802–813. IEEE Computer Society, 2018.

[239] Abdulhamit Subasi and M. Ismail Gursoy. EEG signal classification using PCA, ICA, LDA and support vector machines. *Expert Systems with Applications*, 37(12):8659–8666, 2010.

[240] Zhi-Hao Tan, Teng Zhang, and Wei Wang. Coreset Stochastic Variance-Reduced Gradient with Application to Optimal Margin Distribution Machine. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI, Honolulu, Hawaii, USA, January 27 - February 1*, pages 5083–5090. AAAI Press, 2019.

[241] Murad Tukan, Cenk Baykal, Dan Feldman, and Daniela Rus. On Coresets for Support Vector Machines. In Jianer Chen, Qilong Feng, and Jinhui Xu, editors, *Theory and Applications of Models of Computation, 16th International Conference, TAMC, Changsha, China, October 18-20, Proceedings*, volume 12337 of *Lecture Notes in Computer Science*, pages 287–299. Springer, 2020.

[242] Paxton Turner, Jingbo Liu, and Philippe Rigollet. A Statistical Perspective on Coreset Density Estimation. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS, April 13-15, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2512–2520. PMLR, 2021.

[243] Koen van Greevenbroek. Averaging curves under the dynamic time warping distance. Master's thesis, University of Bonn, July 2020.

[244] Marc J. van Kreveld and Lionov Wiratma. Median trajectories using well-visited regions and shortest paths. In Isabel F. Cruz, Divyakant Agrawal, Christian S. Jensen, Eyal Ofek, and Egemen Tanin, editors, *19th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS, November 1-4, Chicago, IL, USA, Proceedings*, pages 241–250. ACM, 2011.

[245] Vladimir N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[246] Suresh Venkatasubramanian and Qiushi Wang. The Johnson-Lindenstrauss Transform: An Empirical Study. In Matthias Müller-Hannemann and Renato Fonseca F. Werneck, editors, *Proceedings of the Thirteenth Workshop on Algorithm Engineering and Experiments, ALENEX, Holiday Inn San Francisco Golden Gateway, San Francisco, California, USA, January 22*, pages 164–173. SIAM, 2011.

[247] Ingo Wegener. *Complexity Theory: Exploring the Limits of Efficient Algorithms.* Springer Verlag, Berlin, Heidelberg, 2005.

[248] Xunkai Wei and Yinghong Li. Theoretical Analysis of a Rigid Coreset Minimum Enclosing Ball Algorithm for Kernel Regression Estimation. In Fuchun Sun, Jianwei Zhang, Ying Tan, Jinde Cao, and Wen Yu, editors, *Advances in Neural Networks - ISNN, 5th International Symposium on Neural Networks, ISNN, Beijing, China, September 24-28, Proceedings, Part I*, volume 5263 of *Lecture Notes in Computer Science*, pages 741–752. Springer, 2008.

[249] Xunkai Wei, Rob Law, Lei Zhang, Yue Feng, Yan Dong, and Yinghong Li. A fast coreset minimum enclosing ball kernel machines. In *Proceedings of the International Joint Conference on Neural Networks, IJCNN, part of the IEEE World Congress on Computational Intelligence, WCCI, Hong Kong, China, June 1-6*, pages 3366–3373. IEEE, 2008.

[250] Endre Weiszfeld. Sur le point pour lequel la somme des distances de n points donnés est minimum. *Tohoku Mathematical Journal*, 43(2):355–386, 1937.

[251] Endre 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.

[252] Emo Welzl. Smallest enclosing disks (balls and ellipsoids). In Hermann A. Maurer, editor, *New Results and New Trends in Computer Science, Graz, Austria, June 20-21, Proceedings [on occasion of H. Maurer's 50th birthday]*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer, 1991.

[253] E. T. Whittaker and G. N. Watson. *A Course of Modern Analysis*. Cambridge Mathematical Library. Cambridge University Press, 4 edition, 1996.

[254] David P. Woodruff. Optimal space lower bounds for all frequency moments. In J. Ian Munro, editor, *Proceedings of the Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA, New Orleans, Louisiana, USA, January 11-14*, pages 167–175. SIAM, 2004.

[255] Dongkuan Xu and Yingjie Tian. A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science*, 2(2):165–193, June 2015.

[256] Andrew Chi-Chih Yao. Some Complexity Questions Related to Distributive Computing (Preliminary Report). In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, Atlanta, Georgia, USA*, pages 209–213. ACM, 1979.

[257] Chengsheng Yuan, Xingming Sun, and Rui Lv. Fingerprint liveness detection based on multi-scale LPQ and PCA. *China Communications*, 13(7):60–65, 2016.

[258] Jacky Zhang, Rajiv Khanna, Anastasios Kyrillidis, and Sanmi Koyejo. Bayesian Coresets: Revisiting the Nonconvex Optimization Perspective. In Arindam Banerjee and Kenji Fukumizu, editors, *The 24th International Conference on Artificial Intelligence and Statistics, AISTATS, April 13-15, Virtual Event*, volume 130 of *Proceedings of Machine Learning Research*, pages 2782–2790. PMLR, 2021.

[259] Min Zhang, Yinlin Fu, Kevin M. Bennett, and Teresa Wu. Computational efficient Variational Bayesian Gaussian Mixture Models via Coreset. In *International Conference on Computer, Information and Telecommunication Systems, CITS, Kunming, China, July 6-8*, pages 1–5. IEEE, 2016.

[260] Yan Zheng and Jeff M. Phillips. Coresets for Kernel Regression. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17*, pages 645–654. ACM, 2017.

Index