ROBUST ESTIMATION AND SKETCHING OF POINTS, LINES, TRAJECTORIES AND OTHER SHAPES

by

Pingfan Tang

A dissertation submitted to the faculty of
The University of Utah
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy

School of Computing

The University of Utah

June 2019

The University of Utah Graduate School

STATEMENT OF DISSERTATION APPROVAL

The dissertation of **Pingfan Tang**
has been approved by the following supervisory committee members:

Jeff M Phillips, Chair(s) __
Date Approved

Bei Wang Phillips, Member __
Date Approved

Aditya Bhaskara, Member __
Date Approved

Kevin Buchin, Member __
Date Approved

Tom Fletcher, Member __
Date Approved

by **Ross T Whitaker**, Chair/Dean of
the Department/College/School of **Computing**
and by **David B. Kieda**, Dean of The Graduate School.

ABSTRACT

We study robust estimators for uncertain points, and sketching of lines, trajectories and other shapes. For locationally uncertain points, each point in a data set has a discrete probability distribution describing its location. The probabilistic nature of uncertain data makes it challenging to compute such estimators, since the true value of the estimator is now described by a distribution rather than a single point. We show how to construct and estimate the distribution of the median and other robust estimators of an uncertain point set. More generally, for robust estimators, we also give a result about the robustness of composite estimators: under mild conditions on the individual estimators, the breakdown point of the composite estimator is the product of the breakdown points of the individual estimators. Another contribution of this work is a sketched representations based on a set of landmarks for geometric objects. Using this representation, we develop a new class of distances for objects including lines, hyperplanes, and trajectories. These distances easily and interpretably map objects to a Euclidean space, are simple to compute, and perform well in data analysis tasks. For trajectories, they match and in some cases significantly out-perform all state-of-the-art other metrics, can effortlessly be used in k-means clustering, and fast approximate nearest neighbor algorithms which greatly improves the efficiency of trajectory similarity search. Under reasonable and often simple conditions, these distances are metrics. We also show how to use sensitivity sampling to approximate such landmark-based distances, bound the required size of the sketched vector, and give an algorithm to recover a trajectory from its vectorized representation.

For my parents, Yongping Tang and Naijuan Ya.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOTATION AND SYMBOLS

$\mathbb{R}$	the collection of all real numbers
$\mathbb{R}^d$	the d-dimensional Euclidean space
$\mathbb{Z}$	the collection of all integers
$\mathcal{P}$	a set of uncertain points
$\mathcal{H}_p$	the collection of all half spaces that contain the point p
P_{flat}	the union of all uncertain points in $\mathcal{P}$
$\uplus$	the union of multisets
$\mathcal{L}$	the collection of all lines in $\mathbb{R}^2$
$\mathcal{H}$	the collection of all hyperplanes in $\mathbb{R}^d$
$\mathcal{T}$	the collection of all piece-wise linear curves in $\mathbb{R}^2$
$\lVert \cdot \rVert$	Euclidean norm
$\lVert \cdot \rVert_\infty$	l^∞ norm

ACKNOWLEDGEMENT

I would like to express my deepest appreciation to my advisor Professor Jeff M Phillips. He offered me a great opportunity to join the data group in School of Computing. On the academic level, his professional guidance gives me a chance to be exposed to forefront and fascinating research topics in data science. When I encounter difficulties in research, he always provides me with profound encouragement and ingenious suggestions, which are like a lamp helping me overcome obstacles in darkness. Moreover, his strong recommendation is a great help for me to successfully find an internship and a full-time position. In a word, Professor Phillips' constructive advice and relentless support on both my research and career have been invaluable.

Besides my advisor, I would like to acknowledge the assistance of the rest of my dissertation committee members (Bei Wang Phillips, Aditya Bhaskara, Kevin Buchin and Tom Fletcher). Their support and insightful comments in my proposal defense also help me a lot.

CHAPTER 1

INTRODUCTION

With the development of the internet and the world wide web, massive data become more and more commonplace. How to represent, summarize and sketch these data has always been an important research topic in machine learning and data mining. This is because sometimes a good representation of summarization itself can provide us with lots of useful information. For example, combing the median of population's income and living cost, we can have a general understanding to living standard of people in a region. Moreover, a good summarization and sketch can effectively reduce the size of data, which means the data can be easily stored, transmitted and visualized. Most importantly, a good representation or sketch of data can allow us to directly apply many standard machine learning and data mining algorithms, and run these algorithms on small data sets efficiently. This dissertation mainly studies two problems. One is how to analyze and summarize the locationally uncertain data points, especially to model its probabilistic nature. The other is how to effectively sketch lines, trajectories and other geometric shapes, and the property and application of this sketched representation.

For a data set drawn iid (independent and identically distributed) from a single distribution, if we want to use one point to represent this set, then obviously the median is good choice. Since the median is robust to outliers, it is better than the mean in the presence of noise and outliers. However, in the age of big data, a number of indirect data collection methods have led to the proliferation of uncertain data ([1–3]), which can be easily found on the web, in sensor networks, and within enterprises in structured or unstructured sources. For a set of uncertain data, a single point usually cannot give a robust estimate for this set (see the discussion in Section 2.2). To analyze and summarize uncertain data, in Chapter 2, we initiate the study of robust estimators for uncertain data, by studying the median, as well as extensions to the Tukey median and geometric median on locationally uncertain

data points. We show how to efficiently create approximate distributions for the location of these medians in $\mathbb{R}^d$. We also develop a general approximation technique for distributions of robust estimators with respect to ranges with bounded VC dimension (Section 2.4). This includes the geometric median for high dimensions and the Siegel estimator for linear regression.

While studying the robust estimation of uncertain data, we notice the median of the union of all uncertain points is more robust than the median of medians of each uncertain point. This inspires us to study the robustness of composite estimators. The estimator composition usually appears in data analysis pipeline, and is very common in broad data analysis literature. In Chapter 3, we formally define the breakdown point ([32, 52]), introduce the onto-breakdown point and use these conceptions to study the robustness of composite estimators. Generally, the composition of two or more estimators is less robust than each individual estimator.

Another problem we try to address is how to effectively sketch lines, trajectories and general geometric shapes. We introduce sketched representation for geometric objects based on a set of landmarks Q. Each object J is represented by a vector $v_Q(J)$, where each entry $v_q(J)$ is defined as the distance between q and J. Using this vectorized representation, we introduce a new family of landmark-based distances $\mathtt{d}_Q$. For example, the distance between two objects can be simply defined as the (normalized) Euclidean distance between their sketched representations.

In Chapter 4, we give the definition of $\mathtt{d}_Q$ and its variants for lines, hyperplanes and trajectories, and show their nice mathematical properties, e.g., being psuedo-metrics, metrics, and bounded VC-dimension of metric balls. These nice properties allow us to directly apply Algorithm 2.2 in Chapter 2 to robust estimators of linear regression, like Siegel estimator, on uncertain data (Corollary 4.1), and this is also a motivation for us to study $\mathtt{d}_Q$ and sketched representation of geometric shapes. In Chapter 4, we mainly study the application of $\mathtt{d}_Q$ and its variants in trajectories analysis. We apply $\mathtt{d}_Q$ using the KNN classification algorithm to predict trajectory classes from real and synthetic data and compare the result with several other distances to show its competitiveness. Moreover, the vectorized representation also means we can directly apply many existing algorithms in machine learning and data mining to process and analyze geometric objects. For example,

give a set of trajectories we can directly use Lloyd's algorithm to do k-means clustering on it (Section 4.4.2), train SVM classifiers, or run efficient k-nearest neighbor searching algorithms.

In Chapter 5, we study how to approximate $\mathtt{d}_Q$ between two geometric objects when Q is very large. The idea is to use sensitivity sampling ([40, 44, 63, 86]) on Q. For general geometric shapes, we show how to bound the total sensitivity under necessary assumptions. For trajectories, we use the framework in [16] to construct a strong approximation of Q with high probability, and we also give an algorithm to recover a trajectory from its vectorized representation.

1.1 Main Results

Here, we list the main results in this dissertation.

- Given a set of n uncertain points $\mathcal{P}$, where each point has k possible locations, and $\varepsilon \in (0, 1]$, we can construct an ε-approximate coreset T for Tukey median (or L_1 median) on $\mathcal{P}$ that has a size $|T| = O\left(\frac{k^d}{\varepsilon^d}\right)$, and approximately captures the probability of its distribution of its uncertainty (Theorem 2.5, and Theorem 2.8).

- When estimators E_1 and E_2 have breakdown points β_1 and β_2 respectively, we show the general conditions under which an E_1-E_2 estimator has a breakdown point of $\beta_1\beta_2$ (Theorem 3.2), and provide examples when this does not occur.

- In trajectory classification, we show $\mathtt{d}_Q$ and its variants can match and in some cases significantly out-perform all state-of-the-art other distances. We also show the vectorized representation of trajectories can be directly used in k-means clustering, and plugged into approximate nearest neighbor approaches which immediately out-perform the best recent advances in trajectory similarity search by several orders of magnitude (Section 4.4).

- When Q is large, we can use sensitivity sampling to find a set $\tilde{Q} \subset Q$ to approximate $\mathtt{d}_Q$ for pairs of general geometric objects (Theorem 5.6), and construct a strong approximation of Q valid for all trajectories from a mildly restricted family. (Theorem 5.7).

- We design an algorithm which can exactly recover a trajectory γ from a mildly restricted family with k line segments, using only Q and its vectorized representation $v_Q(\gamma)$, in $O(|Q| + k^2)$ time (Theorem 5.8).

CHAPTER 2

APPROXIMATING THE DISTRIBUTION OF THE MEDIAN AND OTHER ROBUST ESTIMATORS ON UNCERTAIN DATA

2.1 Introduction

Most statistical or machine learning models of noisy data start with the assumption that a data set is drawn iid (independent and identically distributed) from a single distribution. Such distributions often represent some true phenomenon under some noisy observation. Therefore, approaches that mitigate the influence of noise, involving robust statistics or regularization, have become commonplace.

However, many modern data sets are clearly not generated iid, rather each data element represents a separate object or a region of a more complex phenomenon. For instance, each data element may represent a distinct person in a population or an hourly temperature reading. Yet, this data can still be noisy; for instance, multiple GPS locational estimates of a person, or multiple temperature sensors in a city. The set of data elements may be noisy *and* there may be multiple inconsistent readings of each element. To model this noise, the inconsistent readings can naturally be interpreted as a probability distribution.

Given such locationally noisy, non-iid data sets, there are many unresolved and important analysis tasks ranging from classification to regression to summarization. In this chapter, we initiate the study of robust estimators [33, 78] on locationally uncertain data. More precisely, we consider an input data set of size n, where each data point's location is described by a discrete probability distribution. We will assume these discrete distributions have a support of at most k points in $\mathbb{R}^d$; and for concreteness and simplicity we will focus on cases where each point has support described by exactly k points, each being equally likely.

Although algorithms for locationally uncertain points have been studied in quite a few

contexts over the last decade [2–5, 28, 55, 60, 67, 94], few have directly addressed the problem of noise in the data. As the uncertainty is often the direct consequence of noise in the data collection process, this is a pressing concern. As such we initiate this study focusing on the most basic robust estimators: the median for data in $\mathbb{R}^1$, as well as its generalization the geometric median and the Tukey median for data in $\mathbb{R}^d$, defined in Section 2.1.1. Being robust refers to the fact that the median and geometric medians have a *breakdown point*s of 0.5, that is, if less than 50% of the data points (the outliers) are moved from the true distribution to some location infinitely far away, the estimator remains within the extent of the true distribution [69]. The Tukey median has a breakdown point between $\frac{1}{d+1}$ and $\frac{1}{3}$ [7].

In this chapter, we generalize the median (and other robust estimators) to locationally uncertain data, where the outliers can occur not just among the n data points, but also as part of the discrete distributions representing their possible locations.

The main challenge is in modeling these robust estimators. As we do not have precise locations of the data, there is not a single minimizer of $\mathsf{cost}(x, Q)$; rather there may be as many as k^n possible input point sets Q (the combination of all possible locations of the data). And the expected value of such a minimizer is not robust in the same way that the mean is not robust. As such we build a distribution over the possible locations of these cost-minimizers. In $\mathbb{R}^1$ (by defining boundary cases carefully) this distribution is of size at most $O(nk)$, the size of the input, but already in $\mathbb{R}^2$ it may be as large as k^n.

Our Results. We design algorithms to create an approximate support of these median distributions. We create small sets T (called an *ε-support*) such that each possible median m_Q from a possible point set Q is within a distance $\varepsilon \cdot \mathsf{cost}(m_Q, Q)$ of some $x \in T$. In $\mathbb{R}$ we can create a support set T of size $O(k/\varepsilon)$ in $O(nk \log(nk))$ time. We show that the bound $O(k/\varepsilon)$ is tight since there may be k large enough modes of these distributions, each requiring $\Omega(1/\varepsilon)$ points to represent. In $\mathbb{R}^d$ our bound on $|T|$ is $O(k^d/\varepsilon^d)$, for the Tukey median and the geometric median. If we do not need to cover sets of medians m_Q which occur with probability less than ε, we can get a bound $O(d/\varepsilon^2)$ in $\mathbb{R}^d$. In fact, this general approach in $\mathbb{R}^d$ extends to other estimators, including the Siegel estimator [81] for linear regression. We then need to map weights onto this support set T. We can do so exactly in $O(n^2k)$ time in $\mathbb{R}^1$ or approximately in $O(1/\varepsilon^2)$ time in $\mathbb{R}^d$.

Another goal may be to then construct a single-point estimator of these distributions: the median of these median distributions. In $\mathbb{R}^1$ we can show that this process is stable up to $\mathsf{cost}(m_Q, Q)$ where m_Q is the resulting single-point estimate. However, we also show that already in $\mathbb{R}^1$ such estimators are not stable with respect to the weights in the median distribution, and hence not stable with respect to the probability of any possible location of an uncertain point. That is, infinitesimal changes to such probabilities can greatly change the location of the single-point estimator. As such, we argue the approximate median distribution (which is stable with respect to these changes) is the best robust representation of such data.

2.1.1 Formalization of Model and Notation

We consider a set of n locationally uncertain points $\mathcal{P} = \{P_1, \ldots, P_n\}$ so that each P_i has k possible locations $\{p_{i,1}, \ldots, p_{i,k}\} \subset \mathbb{R}^d$. Here, $P_i = \{p_{i,1}, \ldots, p_{i,k}\}$ is a multiset, which means a point in P_i may appear more than once. Let $P_{\mathsf{flat}} = \cup_i \{p_{i,1}, \ldots, p_{i,k}\}$ represent all positions of all points in $\mathcal{P}$, which implies P_{flat} is also a multiset. We consider each $p_{i,j}$ to be an equally likely (with probability $1/k$) location of P_i, and can extend our techniques to non-uniform probabilities and uncertain points with fewer than k possible locations. For an uncertain point set $\mathcal{P}$ we say $Q \Subset \mathcal{P}$ is a *traversal* of $\mathcal{P}$ if $Q = \{q_1, \ldots q_n\}$ has each q_i in the domain of P_i (e.g., $q_i = p_{i,j}$ for some j). We denote by $\Pr_{Q \Subset \mathcal{P}}[\gamma(Q)]$ the probability of the event $\gamma(Q)$, given that Q is a randomly selected traversal from $\mathcal{P}$, where the selection of each q_i from P_i is independent of $q_{i'}$ from $P_{i'}$.

We are particularly interested in the case where n is large and k is small. For technical simplicity we assume an extended RAM model where k^n (the number of possible traversals of point sets) can be computed in $O(1)$ time and fits in $O(1)$ words of space.

We consider three definitions of medians. In one dimension, given a set $Q = \{q_1, q_2, \ldots, q_n\}$ that w.l.o.g. satisfies $q_1 \leq q_2 \leq \ldots \leq q_n$, we define the *median* m_Q as $q_{\frac{n+1}{2}}$ when n is odd and $q_{\frac{n}{2}}$ when n is even. There are several ways to generalize the median to higher dimensions [7], herein we focus on the geometric median and Tukey median. Define $\mathsf{cost}(x, Q) = \frac{1}{n}\sum_{i=1}^{n} \|x - q_i\|$ where $\|\cdot\|$ is the Euclidian norm. Given a set $Q = \{q_1, q_2, \ldots, q_n\} \subset \mathbb{R}^d$, the *geometric median* is defined as $m_Q = \arg\min_{x \in \mathbb{R}^d} \mathsf{cost}(x, Q)$. The Tukey depth [83] of a point p with respect to a set $Q \subset \mathbb{R}^d$ is defined $\mathsf{depth}_Q(p) := \min_{H \in \mathcal{H}_p} |H \cap Q|$ where

$\mathcal{H}_p := \{H \text{ is a closed half space in } \mathbb{R}^d \mid p \in H\}$. Then a *Tukey median* of a set Q is a point p that can maximize the Tukey depth.

2.1.2 Related Work on Uncertain Data

The algorithms and computational geometry communities have recently generated a large amount of research in trying to understand how to efficiently process and represent uncertain data [1–5, 28, 55, 60, 62, 67], not to mention some motivating systems and other progress from the database community [6, 30, 31, 79, 94]. Some work in this area considers other models, with either worst-case representations of the data uncertainty [84] which do not naturally allow probabilistic models, or when the data may not exist with some probability [5, 55, 62]. The second model can often be handled as a special case of the locationally uncertain model we study. Among locationally uncertain data, most work focuses on data structures for easy data access [3, 24, 30, 82] but not the direct analysis of data. Among the work on analysis and summarization, such as for histograms [27], convex hulls [5], or clustering [28] it usually focuses on quantities like the expected or most likely value, which may not be stable with respect to noise. This includes estimation of the expected median in a stream of uncertain data [58] or the expected geometric median as part of k-median clustering of uncertain data [28]. We are not aware of any work on modeling the probabilistic nature of locationally uncertain data to construct robust estimators of that data, robust to outliers in both the set of uncertain points as well as the probability distribution of each uncertain point.

2.2 Constructing a Single Point Estimate

We begin by exploring the construction of a single point estimator of set of n locationally uncertain points $\mathcal{P}$. We demonstrate that while the estimator is stable with respect to the value of cost, the actual minimum of that function is not stable and provides an incomplete picture for multimodal uncertainties.

It is easiest to explore this through a weighted point set $X \subset \mathbb{R}^1$. Given a probability distribution defined by $\omega : X \to [0, 1]$, we can compute its weighted median by scanning from smallest to largest until the sum of weights reaches 0.5.

There are two situations whereby we obtain such a discrete weighted domain. The first

domain is the set T of possible locations of medians under different instantiations of the uncertain points with weights $\hat{w}$ as the probability of those medians being realized; see constructions in Section 2.3.2 and Section 2.3.6. Let the resulting weighted median of $(T, \hat{w})$ be m_T. The second domain is simply the set P_{flat} of all possible locations of $\mathcal{P}$, and its weight w where $w(p_{i,j})$ is the fraction of $Q \Subset \mathcal{P}$ which take $p_{i,j}$ as their median (possibly 0). Let the weighted median of (P_{flat}, w) be $m_{\mathcal{P}}$.

Theorem 2.1. $|m_T - m_{\mathcal{P}}| \leq \varepsilon \mathit{cost}(m_{\mathcal{P}}) \leq \varepsilon \mathit{cost}(m_Q, Q)$, *$Q \Subset \mathcal{P}$ is any traversal with $m_{\mathcal{P}}$ as its median.*

Proof. We can divide $\mathbb{R}$ into $|T|$ intervals, one associated with each $x \in T$, as follows. Each $z \in \mathbb{R}$ is in an interval associated with $x \in T$ if z is closer to x than any other point $y \in T$, unless $|z - y| \leq \varepsilon \mathsf{cost}(z)$ but $|z - x| > \mathsf{cost}(z)$. Thus a point $p_{i,j}$ whose weight $w(p_{i,j})$ contributes to $\hat{w}(x)$, is in the interval associated with x.

Thus, if $p_{i,j} = m_{\mathcal{P}}$, then the sum of all weights of all points greater than $p_{i,j}$ is at most 0.5, and the sum of all weights of points less than $p_{i,j}$ is less than 0.5. Hence if $m_{\mathcal{P}}$ is in an interval associated with $x \in T$, then the sum of all weights of points $p_{i,j}$ in intervals greater than that of x must be at most 0.5 and those less than that of x must be less than 0.5. Hence $m_T = x$, and $|x - p_{i,j}| \leq \varepsilon \mathsf{cost}(m_{\mathcal{P}})$ as desired. □

Non-Robustness of single point estimates. The geometric median of the set $\{m_Q$ is a geometric median of $Q \mid Q \Subset \mathcal{P}\}$ is not stable under small perturbations in weights; it stays within the convex hull of the set, but otherwise not much can be said, even in $\mathbb{R}^1$. Consider the example with $n = 3$ and $k = 2$, where $p_{1,1} = p_{1,2} = p_{2,1} = 0$ and $p_{2,2} = p_{3,1} = p_{3,2} = \Delta$ for some arbitrary Δ. The median will be at 0 or Δ, each with probability $1/2$, depending on the location of P_2. We can also create a more intricate example where $\hat{\mathsf{cost}}(0) = \hat{\mathsf{cost}}(\Delta) = 0$. As these examples have m_Q at 0 or Δ equally likely with probability $1/2$, then canonically in $\mathbb{R}^1$ we would have the median of this distribution at 0, but a slight change in probability (say from sampling) could put it all the way at Δ. This indicates that a representation of the distribution of medians as we study in the remainder is more appropriate for noisy data.

2.3 Approximating the Median Distribution

The big challenge in constructing an ε-support T is finding the points $x \in P_{\mathsf{flat}}$ which have small values of $\mathsf{cost}(x, Q)$ (recall $\mathsf{cost}(x, Q) = \frac{1}{n}\sum_{i=1}^{n} \|x - q_i\|$) for some $Q \Subset \mathcal{P}$. But this requires determining the smallest cost $Q \Subset \mathcal{P}$ that has $x \in Q$ and x is the median of Q.

One may think (as the authors initially did) that one could simply use a proxy function $\hat{\mathsf{cost}}(x) = \frac{1}{n}\sum_{i=1}^{n} \min_{1 \le j \le k} \|x - p_{i,j}\|$, which is relatively simple to compute as the lower envelope of cost functions for each P_i. Clearly $\hat{\mathsf{cost}}(x) \le \mathsf{cost}(x, Q)$ for all $Q \Subset \mathcal{P}$, so a set $\hat{T}$ satisfying a similar approximation for $\hat{\mathsf{cost}}$ will satisfy our goals for cost. However, there exist (rather adversarial) data sets $\mathcal{P}$ where $\hat{T}$ would require $\Omega(nk)$ points; see Appendix A.1. On the other hand, we show this is not true for cost. The key difference between cost and $\hat{\mathsf{cost}}$ is that $\hat{\mathsf{cost}}$ does not enforce the use of some $Q \Subset \mathcal{P}$ of which x is a median. That is, that (roughly) half the points are to the left and half to the right for this Q.

*Proxy functions L, R, and D. We handle this problem by first introducing two families of functions, defined precisely shortly. We let $L_i(x)$ (resp. $R_i(x)$) represent the contribution to cost at x from the closest possible location $p_{i,j}$ of an uncertain point P_i to the left (resp. right) of x. This allows us to decompose the elements of this cost. However, it does not help us to enforce this balance. Hence we introduce a third proxy function

$$D_i(x) = L_i(x) - R_i(x)$$

capturing the difference between L_i and R_i. We will show that the choice of which points are used on the left or right of x is completely determined by the D_i values. In particular, we maintain the D_i values (for all $i \in [n]$) in sorted order, and use the i with larger D_i values on the right, and smaller D_i values on the left for the min cost $Q \Subset \mathcal{P}$.

To define L_i, R_i, and D_i, we first assume that P_{flat} and P_i for all $i \in [n]$ are sorted (this would take $O(nk \log(nk))$ time). Then to simplify definitions we add two dummy points to each P_i, and introduce the notation $\widetilde{P}_i = P_i \cup \{p_{i,0}, p_{i,k+1}\}$ and $\widetilde{\mathcal{P}} = \{\widetilde{P}_1, \widetilde{P}_2, \cdots, \widetilde{P}_n\}$, where $p_{i,0} = \min P_{\mathsf{flat}} - n\Delta$, $p_{i,k+1} = \max P_{\mathsf{flat}} + n\Delta$, and $\Delta = \max P_{\mathsf{flat}} - \min P_{\mathsf{flat}}$. Thus, every point $p \in P_{\mathsf{flat}}$ can be viewed as the median of some traversal of $\widetilde{\mathcal{P}}$. Moreover, since we put the $p_{i,0}$ and $p_{i,k+1}$ points far enough out, they will essentially act as points at infinity and not affect the rest of our analysis.

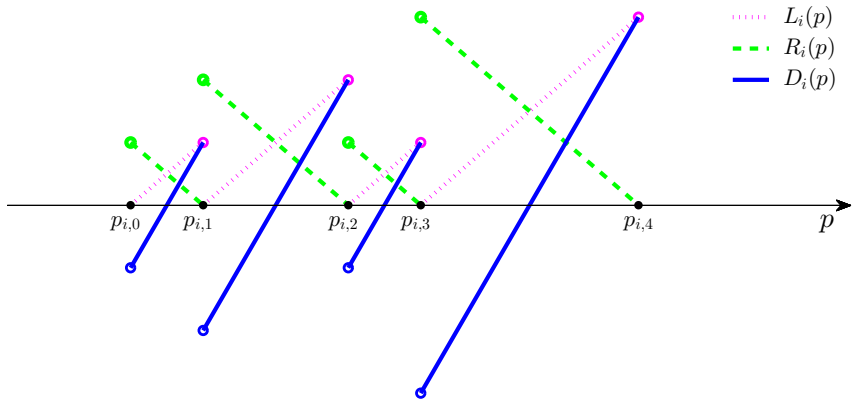

Figure 2.1: The plot of $L_i(p)$, $R_i(p)$ and $D_i(p)$.

Next, for $p \in P_{\mathsf{flat}}$ we define $\mathsf{cost}(p) = \min\{\mathsf{cost}(p, Q) \mid p \text{ is the median of } Q \text{ and } Q \Subset \widetilde{\mathcal{P}}\}$. Thus, if there exists $Q \Subset \mathcal{P}$ such that p is the median of Q, then $\mathsf{cost}(p) \leq \mathsf{cost}(p, Q)$.

Now to compute cost and expedite our analysis, for $p \in [\min P_{\mathsf{flat}} - n\Delta, \max P_{\mathsf{flat}} + n\Delta]$, we define $L_i(p) = \min\{|p_i - p| \mid p_i \in \widetilde{P}_i \cap (-\infty, p]\}$ and $R_i(p) = \min\{|p_i - p| \mid p_i \in \widetilde{P}_i \cap [p, \infty)\}$. and recall $D_i(p) = L_i(p) - R_i(p)$. Obviously, if $p \in \widetilde{P}_i$, then $D_i(p) = L_i(p) = R_i(p) = 0$. For example, if $\widetilde{P}_i = \{p_{i,0},\ p_{i,1},\ p_{i,2},\ p_{i,3},\ p_{i,4}\}$ and $p_{i,0} < p_{i,1} < p_{i,2} < p_{i,3} < p_{i,4}$, then the plot of $L_i(p)$, $R_i(p)$ and $D_i(p)$, is shown in Figure 2.1.

For the sake of brevity, we now assume n is odd; adjusting a few arguments by $+1$ will adjust for the n is even case.

Consider next the following property of the D_i functions with respect to computing $\mathsf{cost}(p)$ for a point $p \in P_{i_0}$. Let $\{i_1, i_2, \cdots, i_{n-1}\} = [n] \backslash \{i_0\}$ be a permutation of uncertain points, except for i_0, so that $D_{i_1}(p) \leq D_{i_2}(p) \leq \cdots \leq D_{i_{n-1}}(p)$. Then to minimize $\mathsf{cost}(p, Q)$, we count the uncertain points P_{i_l} using L_{i_l} if in the permutation $i_l \leq (n-1)/2$ and otherwise count it on the right with R_{i_l}. This holds since for any other permutation $\{j_1, j_2, \cdots, j_{n-1}\} = [n] \backslash \{i_0\}$ we have $\sum_{l=\frac{n+1}{2}}^{n-1} D_{i_l}(p) \geq \sum_{l=\frac{n+1}{2}}^{n-1} D_{j_l}(p)$ and thus

$$\begin{aligned}\sum_{l=1}^{\frac{n-1}{2}} L_{i_l}(p) + \sum_{l=\frac{n+1}{2}}^{n-1} R_{i_l}(p) &= \sum_{l=1}^{n-1} L_{i_l}(p) - \sum_{l=\frac{n+1}{2}}^{n-1} D_{i_l}(p) \\ &\leq \sum_{l=1}^{n-1} L_{j_l}(p) - \sum_{l=\frac{n+1}{2}}^{n-1} D_{j_l}(p) = \sum_{l=1}^{\frac{n-1}{2}} L_{j_l}(p) + \sum_{l=\frac{n+1}{2}}^{n-1} R_{j_l}(p).\end{aligned}$$

For $p \in P_{i_0}$, $\mathsf{cost}(p) = \frac{1}{n}\big(\sum_{l=1}^{\frac{n-1}{2}} L_{i_l}(p) + \sum_{l=\frac{n+1}{2}}^{n-1} R_{i_l}(p)\big)$ under this D_i-sorted permutation.

2.3.1 Computing cost

Now to compute cost for all points $p \in P_{\mathsf{flat}}$, we simply need to maintain the D_i in sorted order, and then sum the appropriate terms from L_i and R_i. Let us first examine a few facts about the complexity of these functions.

The function L_i (resp. R_i) is piecewise-linear, where the slope is always 1 (resp. -1). The breakpoints only occur at $x = p_{i,j}$ for each $p_{i,j} \in P_i$. Hence, they each have complexity $\Theta(k)$ for all $i \in [n]$. The structure of L_i and R_i implies that D_i is also piecewise-linear, where the slope is always 2 and has breakpoints for each $p_{i,j} \in P_i$. Each linear component attains a value $D_i(x) = 0$ when x is the midpoint between two $p_{i,j}, p_{i,j'} \in P_i$ which are consecutive in the sorted order of P_i.

The fact that all D_i have slope 2 at all non-discontinuous points, and these discontinuous points only occur at P_i, implies that the sorted order of the D_i functions does not change in between points of P_{flat}. Moreover, at one of these points of discontinuity $x \in P_{\mathsf{flat}}$, the ordering between D_is only changes for uncertain points $D_{i'}$ such that there exists a possible location $p_{i',j} \in P_{i'}$ such that $x = p_{i',j}$. This implies that to maintain the sorted order of D_i for any x, as we increase the value of x, we only need to update this order at the nk points in P_{flat} with respect to $D_{i'}$ for which there exists $p_{i',j} \in P_{i'}$ with $p_{i',j} = x$. This takes $O(\log(nk))$ time per update using a balanced BST, and thus $O(nk \log(nk))$ time to define $\mathsf{cost}(x)$ for all values $x \in \mathbb{R}^1$. To compute $\mathsf{cost}(x)$, we also require the values of L_i (or R_i); these can be constructed independently for each $i \in [n]$ in $O(k)$ time after sorting, and in $O(nk \log k)$ time overall.[1] Ultimately, we arrive at the following theorem.

Theorem 2.2. *Consider a set of n uncertain points $\mathcal{P}$ with k possible locations each. We can compute $\mathsf{cost}(x)$ for all $x \in \mathbb{R}$ such that $x = p_{i,j}$ for some $p_{i,j} \in P_{\mathsf{flat}}$ in $O(nk \log(nk))$ time.*

2.3.2 Building the ε-Support T and Bounding its Size

We next show that there always exists an ε-support T and it has a size $|T| = O(\frac{k}{\varepsilon})$.

---

[1]When multiple distinct $p_{i,j}$ coincide at a point x, then more care may be required to compute $\mathsf{cost}(x)$ (depending on the specifics of how the median is defined in these boundary cases). Specifically, we may not want to set $L_i(x) = 0$, instead it may be better to use the value $R_i(x)$ even if $R_i(x) = \alpha > 0$. This is the case when $\alpha < R_{i'}(x) - L_{i'}(x)$ for some other uncertain point $P_{i'}$ (then we say P_i is on the right, and P_i is on the left). This can be resolved by either tweaking the definition of median for these cases, or sorting all $D_i(x)$ for uncertain points P_i with some $p_{i,j} = x$, and some bookkeeping.

Theorem 2.3. *Given a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}$, and $\varepsilon \in (0, 1]$ we can construct an ε-support T that has a size $|T| = O(\frac{k}{\varepsilon})$.*

Proof. We first sort P_{flat} in ascending order, scan $P_{\mathsf{flat}} = \{p_1, \cdots, p_{nk}\}$ from left to right and choose one point from P_{flat} every $\lfloor \frac{n}{3} \rfloor$ points, and then put the chosen point into T. Now, suppose p is the median of some traversal $Q \Subset \mathcal{P}$ and $\mathsf{cost}(p) = \mathsf{cost}(p, Q)$. If $p \notin T$, then there are two consecutive points t, t' in T such that $t < p < t'$. On either side of p there are at least $\lfloor \frac{n}{2} \rfloor$ points in Q, so without loss of generality, we assume $|p - t'| \geq \frac{1}{2}|t - t'|$. Since $|[p, \infty) \cap Q| \geq \frac{n}{2}$ and there are at most $\lfloor \frac{n}{3} \rfloor$ points in $[p, t']$, we have $|(t', \infty) \cap Q| \geq \frac{n}{2} - \lfloor \frac{n}{3} \rfloor \geq \frac{n}{6}$, which implies

$$\begin{aligned} \mathsf{cost}(p) =& \mathsf{cost}(p, Q) \geq \frac{1}{n} \sum_{q \in (t', \infty) \cap Q} |q - p| \geq \frac{1}{n} \sum_{q \in (t', \infty) \cap Q} |t' - p| \\ \geq& \frac{1}{n} \frac{n}{6} |t' - p| = \frac{1}{6} |t' - p| \geq \frac{1}{12} |t - t'|. \end{aligned} \tag{2.1}$$

For any fixed $\varepsilon \in (0, 1]$, and two consecutive points t, t' $(t < t')$ in T, we put $x_1, \cdots, x_{\lceil \frac{12}{\varepsilon} \rceil - 1}$ into T where $x_i = t + \frac{|t - t'| i}{\lceil \frac{12}{\varepsilon} \rceil}$ for $1 \leq i \leq \lceil \frac{12}{\varepsilon} \rceil - 1$. So, for the median $p \in (t, t')$, there exists $x_i \in T$ s.t. $|p - x_i| \leq \frac{\varepsilon}{12} |t - t'|$, and from (2.1), we know $|p - x_i| \leq \varepsilon \mathsf{cost}(p)$. In total we put $O(\frac{k}{\varepsilon})$ points into T; thus the proof is completed. □

Remark 2.1. *The above construction results in an ε-support T of size $O(k/\varepsilon)$, but does not restrict that $T \subset P_{\mathsf{flat}}$. We can enforce this restriction by for each x placed in T to choose the single nearest point $p \in P_{\mathsf{flat}}$ to replace it in T. This results in an (2ε)-support, which can be made an ε-support by instead adding $\lceil \frac{24}{\varepsilon} \rceil - 1$ points between each pair (t, t'), without affecting the asymptotic time bound.*

Remark 2.2. *We can construct a sequence of uncertain data $\{\mathcal{P}(n, k)\}$ such that, for each uncertain data $\mathcal{P}(n, k)$, the optimal ε-support T has a size $\Omega(\frac{k}{\varepsilon})$. For example, for $\varepsilon = \frac{1}{3}, \frac{1}{5}, \frac{1}{7}, \cdots$, we define $n = \frac{1}{\varepsilon}$, and $p_{i,j} = (j - 1)n + i$ for $i \in [n]$ and $j \in [k]$. Then, for any median $p \in P_{\mathsf{flat}}$, we have $\varepsilon \mathsf{cost}(p) = \frac{2}{n^2} \sum_{i=1}^{\frac{n-1}{2}} i = \frac{n^2 - 1}{4n^2} < \frac{1}{4}$, hence covering no other points, which implies $|T| = \Omega(nk) = \Omega(\frac{k}{\varepsilon})$.*

We can construct the minimal size ε-support T in $O(nk \log(nk))$ time by sorting, and greedily adding the smallest point not yet covered each step. This yields the slightly stronger corollary of Theorem 2.3.

Corollary 2.1. *Consider a set of n uncertain points* $\mathcal{P} = \{P_1, \cdots, P_n\}$*, where* $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}$*, and* $\varepsilon \in (0, 1]$*. We can construct an* ε*-support* T *in* $O(nk \log(nk))$ *time which has the minimal size for any* ε*-support, and* $|T| = O(\frac{k}{\varepsilon})$*.*

There are multiple ways to generalize the notion of a median to higher dimensions [7]. We focus on two variants: the Tukey median and the geometric median. We start with generalizing the notion of an ε-support to a Tukey median since it more directly follows from the techniques in Theorem 2.3, and then address the geometric median.

2.3.3 An ε-Support for the Tukey Median

A closely related concept to the Tukey median is a *centerpoint*, which is a point p such that $\mathsf{depth}_Q(p) \geq \frac{1}{d+1}|Q|$. Since for any finite set $Q \in \mathbb{R}^d$ its centerpoint always exists, a Tukey median must be a centerpoint. This means if p is the Tukey median of Q, then for any closed half space containing p, it contains at least $\frac{1}{d+1}|Q|$ points of Q. Using this property, we can prove the following theorem.

Theorem 2.4. *Given a set of n uncertain points* $\mathcal{P} = \{P_1, \cdots, P_n\}$*, where* $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$*, and* $\varepsilon \in (0, 1]$*, we can construct an* ε*-support* T *for the Tukey median on* $\mathcal{P}$ *that has a size* $|T| = O(\frac{k^2}{\varepsilon^2})$*.*

Proof. Suppose the projections of P_{flat} on x-axis and y-axis are X and Y respectively. We sort all points in X and choose one point from X every $\lfloor \frac{n}{4} \rfloor$ points, and then put the chosen points into a set X_T. For each point $x \in X_T$ we draw a line through $(x, 0)$ parallel to y-axis. Similarly, we sort all points in Y and choose one point every $\lfloor \frac{n}{4} \rfloor$ points, and put the chosen points into Y_T. For each point $y \in Y_T$ we draw a line through $(0, y)$ parallel to x-axis.

Now, suppose p with coordinates (x_p, y_p) is the Tukey median of some traversal $Q \Subset \mathcal{P}$ and $\mathsf{cost}(p, Q) = \frac{1}{n}\sum_{q \in Q} \|q - p\|$. If $x_p \notin X_T$ and $y_p \notin Y_T$, then there are $x, x' \in X_T$ and $y, y' \in Y_T$ such that $x < x_p < x'$ and $y < y_p < y'$, as shown in Figure 2.2(Left).

Without loss of generality, we assume $|x_p - x| \geq \frac{1}{2}|x' - x|$ and $|y_p - y| \geq \frac{1}{2}|y - y'|$. Since p is the Tukey median of Q, we have $|Q \cap (-\infty, \infty) \times (-\infty, y_p]| \geq \frac{n}{3}$ where $(-\infty, \infty) \times (-\infty, y_p] = \{(x, y) \in \mathbb{R}^2 |\ y \leq y_p\}$. Recall there are at most $\lfloor \frac{n}{4} \rfloor$ points of P_{flat} in $(-\infty, \infty) \times [y_p, y]$, which implies $|Q \cap (-\infty, \infty) \times (-\infty, y)| \geq \frac{n}{3} - \lfloor \frac{n}{4} \rfloor \geq \frac{n}{12}$. So, we have

$$\mathsf{cost}(p,Q) \geq \frac{1}{n}\sum_{q\in Q\cap(-\infty,\infty)\times(-\infty,y)} \|q-p\| \geq \frac{1}{n}\frac{n}{12}|y-y_p| \geq \frac{1}{24}|y-y'|.$$

Using a symmetric argument, we can obtain $\mathsf{cost}(p,Q) \geq \frac{1}{24}|x-x'|$.

For any fixed $\varepsilon \in (0,1]$, and any two consecutive points x, x' in X_T we put $x_1, \cdots, x_{\lceil \frac{48}{\varepsilon} \rceil - 1}$ into X_T where $x_i = x + \frac{|x-x'|i}{\lceil \frac{48}{\varepsilon} \rceil}$. Also, for any two consecutive point y, y' in Y_T, we put $y_1, \cdots, y_{\lceil \frac{48}{\varepsilon} \rceil - 1}$ into Y_T where $y_i = y + \frac{|y-y'|i}{\lceil \frac{48}{\varepsilon} \rceil}$. So, for the Tukey median $p \in (x,x') \times (y,y')$, there exist $x_i \in X_T$ and $y_j \in Y_T$ such that $|x_p - x_i| \leq \frac{\varepsilon}{48}|x-x'|$ and $|y_p - y_j| \leq \frac{\varepsilon}{48}|y-y'|$. Since we have shown that $\frac{1}{24}|x-x'|$ and $\frac{1}{24}|y-y'|$ are lower bounds for $\mathsf{cost}(p,Q)$, we obtain

$$\begin{aligned}\|(x_p,y_p)-(x_i,y_j)\| \leq & |x_p - x_i| + |y_p - y_j| \leq \frac{\varepsilon}{48}(|x-x'| + |y-y'|) \\ \leq & \frac{\varepsilon}{48}(24\mathsf{cost}(p,Q) + 24\mathsf{cost}(p,Q)) = \varepsilon\mathsf{cost}(p,Q).\end{aligned}$$

Finally, we define T as $T := X_T \times Y_T$. Then for any $Q \Subset \mathcal{P}$, if p is the Tukey median of Q, there exists $t \in T$ such that $\|t-p\| \leq \varepsilon\mathsf{cost}(p,Q)$. Thus, T is an ε-support for the Tukey median on $\mathcal{P}$. Moreover, since $|X_T| = O(\frac{k}{\varepsilon})$ and $|Y_T| = O(\frac{k}{\varepsilon})$, we have $|T| = O(\frac{k^2}{\varepsilon^2})$. □

In a straight-forward extension, we can generalize the result of Theorem 2.4 to d dimensions.

Theorem 2.5. *Given a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^d$, and $\varepsilon \in (0,1]$, we can construct an ε-support T for the Tukey median on $\mathcal{P}$ that has a size $|T| = O((2d(d+1)(d+2)^2\frac{k}{\varepsilon})^d)$.*

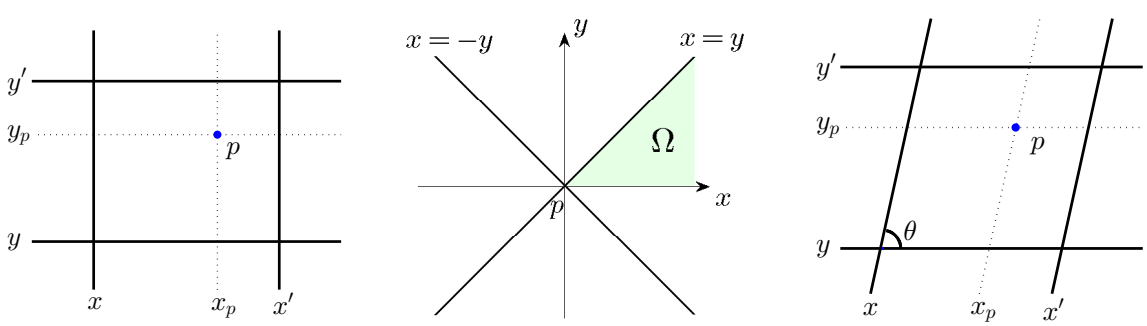

Figure 2.2: Left: Tukey median p is in a grid cell formed by x, x' and y, y'. Center: The plane is decomposed into 8 regions with the same shape. Right: Geometric median p is in an oblique grid cell formed by x, x' and y, y'.

2.3.4 An ε-Support for the Geometric Median

Unlike the Tukey median, there does not exist a constant $C > 0$ such that: for any geometric median p of point set $Q \subset \mathbb{R}^d$, any closed half space containing p contains at least $\frac{1}{C}|Q|$ points of Q. For example, suppose in $\mathbb{R}^2$ there are $2n+1$ points on x-axis with the median point at the origin; this point is also the geometric median. If we move this point upward along the y direction, then the geometric median also moves upwards. However, for the line through the new geometric median and parallel to the x-axis, all $2n$ other points are under this line.

Hence, we need a new idea to adapt the method in Theorem 2.5 for the geometric median in $\mathbb{R}^d$. We first consider the geometric median in $\mathbb{R}^2$. We show we can find *some* line through it, such that on both sides of this line there are at least $\frac{n}{8}$ points.

Lemma 2.1. *Suppose p is the geometric median of $Q \subset \mathbb{R}^2$ with size $|Q| = n$. There is a line ℓ through p so both closed half planes with ℓ as boundary contain at least $\frac{n}{8}$ points of Q.*

Proof. We first build a rectangular coordinate system at the point p, which means p is the origin with coordinates $(x_p, y_p) = (0,0)$. Then we use the x-axis, y-axis and lines $x = y$, $x = -y$ to decompose the plane into eight regions, as shown in Figure 2.2(Center). Since all these eight regions have the same shape, without loss of generality, we can assume $\Omega = \{(x,y) \in \mathbb{R}^2 |\ x \geq y \geq 0\}$ contains the most points of Q. Then $|\Omega \cap Q| \geq \frac{n}{8}$, otherwise $n = |Q| = |\mathbb{R}^2 \cap Q| \leq 8|\Omega \cap Q| < n$, which is a contradiction.

If $|Q \cap \{p\}| \geq \frac{n}{8}$, i.e., the multiset Q contains p at least $\frac{n}{8}$ times, then obviously this proposition is correct. So, we only need to consider the case $|Q \cap \{p\}| < \frac{n}{8}$. We introduce notations $\widetilde{\Omega} = \Omega \setminus \{p\}$ and $\Omega^o = \Omega \setminus \partial\Omega$, and denote the coordinates of any $q \in Q$ as $q = (x_q, y_q)$. From a property of the geometric median (proven in Appendix A.1) we know $\sum_{q \in Q \setminus \{p\}} \frac{x_q - x_p}{\|q-p\|} \leq |Q \cap \{p\}|$. Since $|Q \cap \{p\}| < \frac{n}{8}$ this implies

$$\sum\nolimits_{q \in Q \cap \widetilde{\Omega}} \frac{x_q}{\|q\|} + \sum\nolimits_{q \in Q \setminus \Omega} \frac{x_q}{\|q\|} < \frac{n}{8},$$

since p is the origin and $Q \setminus \{p\} = (Q \cap \widetilde{\Omega}) \cup (Q \setminus \Omega)$. From $\frac{x_q}{\|q\|} = \frac{x_q}{\sqrt{x_q^2 + y_q^2}} \geq \frac{1}{\sqrt{2}}, \forall q \in \widetilde{\Omega}$ we obtain

$$|Q \cap \widetilde{\Omega}| \frac{1}{\sqrt{2}} \leq \sum\nolimits_{q \in Q \cap \widetilde{\Omega}} \frac{x_q}{\|q\|} < \frac{n}{8} - \sum\nolimits_{q \in Q \setminus \Omega} \frac{x_q}{\|q\|} \leq \frac{n}{8} + |Q \setminus \Omega| \leq \frac{n}{8} + (n - |Q \cap \widetilde{\Omega}|)$$

which implies there are not too many points in $\widetilde{\Omega}$,

$$|Q \cap \widetilde{\Omega}| < \frac{\sqrt{2}n}{(1+\sqrt{2})} \cdot \frac{9}{8} < 0.66n.$$

Now, we define the two pairs of half spaces which share a boundary with $\widetilde{\Omega}$: $H_1^+ = \{(x,y) \in \mathbb{R}^2 | \ y \geq 0\}$, $H_1^- = \{(x,y) \in \mathbb{R}^2 | \ y \leq 0\}$ and $H_2^+ = \{(x,y) \in \mathbb{R}^2 | \ x - y \geq 0\}$, $H_2^- = \{(x,y) \in \mathbb{R}^2 | \ x - y \leq 0\}$. We assert either $|H_1^+ \cap Q| \geq \frac{n}{8}$ and $|H_1^- \cap Q| \geq \frac{n}{8}$, or $|H_2^+ \cap Q| \geq \frac{n}{8}$ and $|H_2^- \cap Q| \geq \frac{n}{8}$. Otherwise, since $|Q \cap \Omega| \geq \frac{n}{8}$ and $\Omega \subset H_1^+ \cap H_2^+$, we have $|H_1^- \cap Q| < \frac{n}{8}$ and $|H_2^- \cap Q| < \frac{n}{8}$. From $H_1^- \cup H_2^- \cup \Omega^o = \mathbb{R}^2$ we have

$$\begin{aligned} n =& |Q| = |\mathbb{R}^2 \cap Q| = |(H_1^- \cup H_2^- \cup \Omega^o) \cap Q| \leq |H_1^- \cap Q| + |H_2^- \cap Q| + |\Omega^o \cap Q| \\ \leq& |H_1^- \cap Q| + |H_2^- \cap Q| + |\widetilde{\Omega} \cap Q| \leq \frac{n}{8} + \frac{n}{8} + 0.66n < n, \end{aligned}$$

which is a contradiction. Therefore, among lines $\ell_1 : y = 0$ and $\ell_2 : x - y = 0$, which both go through p, one of them has at least $n/8$ points from Q on both sides. □

Theorem 2.6. *Given a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$, and $\varepsilon \in (0,1]$, we can construct an ε-support T for the geometric median on $\mathcal{P}$ that has a size $|T| = O(\frac{k^2}{\varepsilon^2})$.*

Proof. The idea to prove this theorem is to use several oblique coordinate systems. We consider an oblique coordinate system, the angle between x-axis and y-axis is $\theta \in (0, \frac{\pi}{2}]$, and use the technique in Theorem 2.4 to generate a grid. More precisely, we project P_{flat} onto the x-axis along the y-axis of the oblique coordinate system to obtain a set X, sort all points in X, and choose one point from X every $\lfloor \frac{n}{9} \rfloor$ points to form a set X_T. Then we use the same method to generate Y and Y_T projecting along the x-axis in the oblique coordinate system. For each point $x \in X_T$ we draw a line through $(x, 0)$ parallel to the (oblique) y-axis, and for each point $y \in Y_T$ we draw a line through $(0, y)$ parallel to the (oblique) x-axis.

Let p with coordinates (x_p, y_p) be the geometric median of some traversal $Q \Subset \mathcal{P}$ and $\mathsf{cost}(p, Q) = \frac{1}{n} \sum_{q \in Q} \|q - p\|$. If $x_p \notin X_T$ and $y_p \notin Y_T$, then there are $x, x' \in X_T$ and $y, y' \in Y_T$ such that $x_p \in (x, x')$ and $y_p \in (y, y')$, as shown in Figure 2.2(Right).

If we have the condition:

$$\begin{aligned} |Q \cap (-\infty, \infty) \times (-\infty, y_p]| \geq \frac{n}{8}, \ \ |Q \cap (-\infty, \infty) \times [y_p, \infty)| \geq \frac{n}{8}, \\ |Q \cap (-\infty, x_p] \times (-\infty, \infty)| \geq \frac{n}{8}, \ \ |Q \cap [x_p, \infty) \times (-\infty, \infty)| \geq \frac{n}{8}, \end{aligned} \tag{2.2}$$

then we can make the following computation.

Without loss of generality, we assume $|x_p - x| \geq \frac{1}{2}|x' - x|$ and $|y_p - y| \geq \frac{1}{2}|y - y'|$. There are at most $\lfloor \frac{n}{9} \rfloor$ points of P_{flat} in $(-\infty, \infty) \times [y_p, y]$, which implies $|Q \cap (-\infty, \infty) \times (-\infty, y)| \geq \frac{n}{8} - \lfloor \frac{n}{9} \rfloor \geq \frac{n}{72}$. So, we have

$$\mathsf{cost}(p, Q) \geq \frac{1}{n} \sum\nolimits_{q \in Q \cap (-\infty,\infty) \times (-\infty, y)} \|q - p\| \geq \frac{1}{n} \frac{n}{72} |y - y_p| \geq \frac{\sin(\theta)}{144} |y - y'|.$$

Similarly, we can prove $\mathsf{cost}(p, Q) \geq \frac{\sin(\theta)}{144}|x - x'|$.

For any fixed $\varepsilon \in (0, 1]$, and any two consecutive points x, x' in X_T we put $x_1, \cdots, x_{\lceil \frac{288}{\varepsilon \sin(\theta)} \rceil - 1}$ into X_T where $x_i = x + \frac{|x - x'| i}{\lceil \frac{288}{\varepsilon \sin(\theta)} \rceil}$. Also, for any two consecutive point y, y' in Y_T, we put $y_1, \cdots, y_{\lceil \frac{288}{\varepsilon \sin(\theta)} \rceil - 1}$ into Y_T where $y_i = y + \frac{|y - y'| i}{\lceil \frac{288}{\varepsilon \sin(\theta)} \rceil}$. So, for the L_1 median $p \in (x, x') \times (y, y')$, there exist $x_i \in X_T$ and $y_j \in Y_T$ such that $|x_p - x_i| \leq \frac{\varepsilon \sin(\theta)}{288}|x - x'|$ and $|y_p - y_j| \leq \frac{\varepsilon \sin(\theta)}{288}|y - y'|$. Since we have shown that both $\frac{\sin(\theta)}{144}|x - x'|$ and $\frac{\sin(\theta)}{144}|y - y'|$ are lower bounds for $\mathsf{cost}(p, Q)$, using the distance formula in an oblique coordinate system, we have

$$\begin{aligned}
\|(x_p, y_p) - (x_i, y_j)\| \leq & ((x_p - x_i)^2 + (y_p - y_j)^2 + 2(x_p - x_i)(y_i - y_p)\cos(\theta))^{\frac{1}{2}} \\
\leq & ((x_p - x_i)^2 + (y_p - y_j)^2 + 2|x_p - x_i||y_i - y_p|)^{\frac{1}{2}} = |x_p - x_i| + |y_p - y_j| \\
\leq & \frac{\varepsilon \sin(\theta)}{288}(|x - x'| + |y - y'|) \\
\leq & \frac{\varepsilon \sin(\theta)}{288}\left(\frac{144}{\sin(\theta)}\mathsf{cost}(p, Q) + \frac{144}{\sin(\theta)}\mathsf{cost}(p, Q)\right) = \varepsilon \mathsf{cost}(p, Q).
\end{aligned}$$

Therefore, if all k^n geometric medians of traversals satisfy (2.2) and $\theta \in (0, \frac{\pi}{2}]$ is a constant then $T = X_T \times Y_T$ is an ε-support of size $O\left(\frac{k^2}{(\sin(\theta)\varepsilon)^2}\right)$ for the geometric median on $\mathcal{P}$.

Although we cannot find an oblique coordinate system to make (2.2) hold for all k^n medians, we can use several oblique coordinate systems. Using the result of Lemma 2.1, for any geometric median of n points Q, we know there exists a line ℓ through p and parallel to a line in $\{\ell_1 : y = 0,\ \ell_2 : x - y = 0,\ \ell_3 : x = 0,\ \ell_4 : x + y = 0\}$, such that in both sides of this line, there are at least $\frac{n}{8}$ points of Q. Since we did not make any assumption on the distribution of points in Q, if we rotate $\ell_1, \ell_2, \ell_3, \ell_4$ anticlockwise by $\frac{\pi}{8}$ around the origin, we can obtain four lines $\ell'_1, \ell'_2, \ell'_3, \ell'_4$, and there exists a line ℓ' through p and parallel to a line in $\{\ell'_1, \ell'_2, \ell'_3, \ell'_4\}$, such that on both sides of this line, there are at least $\frac{n}{8}$ points of Q. The angle between ℓ and ℓ' is at least $\frac{\pi}{8}$.

Therefore, given $\mathcal{L} = \{\ell_1, \ell_2, \ell_3, \ell_4\}$ and $\mathcal{L}' = \{\ell'_1, \ell'_2, \ell'_3, \ell'_4\}$, for each pair $(\ell, \ell') \in \mathcal{L} \times \mathcal{L}'$, we take ℓ and ℓ' as x-axis and y-axis respectively to build an oblique coordinate system,

and then use the above method to compute a set $T(\ell, \ell')$. Since for any geometric median p there must be an oblique coordinate system based on some $(\ell, \ell') \in \mathcal{L} \times \mathcal{L}'$ to make (2.2) hold for p, we can take $T = \cup_{\ell \in \mathcal{L}, \ell' \in \mathcal{L}'} T(\ell, \ell')$ as an ε-support for geometric median on $\mathcal{P}$, and the size of T is $|T| = O\left(16 \frac{k^2}{(\sin(\frac{\pi}{8})\varepsilon)^2}\right) = O\left(\frac{k^2}{\varepsilon^2}\right)$. □

2.3.5 Size bound of T in $\mathbb{R}^d$

Using the method in the proof of Theorem 2.6, we can generalize the result of this theorem to $\mathbb{R}^3$ and higher dimensional space.

Theorem 2.7. *Given a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^3$, and $\varepsilon \in (0, 1]$, we can construct an ε-support T for L_1 median on $\mathcal{P}$ that has a size $|T| = O\left(\frac{k^3}{\varepsilon^3}\right)$.*

Proof. The first step is to obtain a result similar to Lemma 2.1: if p is the L_1 median of a set of n points $Q \subset \mathbb{R}^3$, then we can find a plane h through p, such that any closed half space with h as its boundary contains at least $\frac{n}{24}$ points of Q.

To prove this, we build a rectangular coordinate system at the point p, and use nine planes $\mathcal{H}_3 = \{x_1 = 0, x_2 = 0, x_3 = 0, x_1 \pm x_2 = 0, x_2 \pm x_3 = 0, x_3 \pm x_1 = 0\}$ to partition $\mathbb{R}^3$ into 24 regions: $\{\Omega_{i,\mathbf{s}} |\ i \in \{1, 2, 3\}, \mathbf{s} \in \{1, -1\}^3\}$, where $\Omega_{i,\mathbf{s}} = \Omega_{i,(s_1,s_2,s_3)} := \{(x_1, x_2, x_3) \in \mathbb{R}^3 |\ s_i x_i \geq s_j x_j \geq 0,\ \text{for } j = 1, 2, 3\}$. All of these regions have the same shape with $\Omega_{1,(1,1,1)} = \{(x_1, x_2, x_3) \in \mathbb{R}^3 | x_1 \geq x_2 \geq 0, x_1 \geq x_3 \geq 0\}$, which means they can coincide with each other through rotation, shift and reflection. So, we define $\Omega = \Omega_{1,(1,1,1)}$ and without loss of generality assume $|Q \cap \Omega| = \max_{i \in [3], \mathbf{s} \in \{1,-1\}^3} |Q \cap \Omega_{i,\mathbf{s}}|$. Obviously, we have $|Q \cap \Omega| \geq \frac{n}{24}$.

We only need to consider the case $|Q \cap \{p\}| < \frac{n}{24}$. Introducing notations $\widetilde{\Omega} = \Omega \setminus \{p\}$, $\Omega^o = \Omega \setminus \partial\Omega$, from the property of L_1 median we know $\sum_{q \in Q \setminus \{p\}} \frac{x_{q,1} - x_{p,1}}{\|q - p\|} \leq |Q \cap \{p\}| < \frac{n}{24}$. Since p is the origin, we have$\sum_{q \in Q \cap \widetilde{\Omega}} \frac{x_{q,1}}{\|q\|} + \sum_{q \in Q \setminus \Omega} \frac{x_{q,1}}{\|q\|} < \frac{n}{24}$, which implies $|Q \cap \widetilde{\Omega}| \frac{1}{\sqrt{3}} < \frac{n}{24} + |Q \setminus \Omega| \leq \frac{n}{24} + (n - |Q \cap \widetilde{\Omega}|)$ since $\frac{x_{q,1}}{\|q\|} \leq \frac{1}{\sqrt{3}}$, for all $q \in \widetilde{\Omega}$. Thus, we obtain

$$|Q \cap \widetilde{\Omega}| < \frac{\sqrt{3}n}{1 + \sqrt{3}} \cdot \frac{25}{24} < 0.67n. \tag{2.3}$$

Now, for $x = (x_1, x_2, x_3) \in \mathbb{R}^3$ we define $h_1(x) = x_1 - x_2$, $h_2(x) = x_1 - x_3$, $h_3(x) = x_2$, $h_4(x) = x_3$, and $H_i^+ = \{x \in \mathbb{R}^3 |\ h_i(x) \geq 0\}$, $H_i^- - = \{x \in \mathbb{R}^3 |\ h_i(x) \leq 0\}$, and assert there exists $i \in [4]$ such that $|H_i^+ \cap Q| \geq \frac{n}{24}$ and $|H_i^- \cap Q| \geq \frac{n}{24}$. Otherwise, since $|Q \cap \Omega| \geq \frac{n}{24}$

and $\Omega \subset \cap_{i=1}^{4} H_i^+$, we have $|H_i^- \cap Q| < \frac{n}{24}$ for all $i \in [4]$. From $\cup_{i=1}^{4} H_i^- \cup \Omega^o = \mathbb{R}^3$ and (2.3) we have

$$\begin{aligned} n =& |Q| = |\mathbb{R}^3 \cap Q| = |(\cup_{i=1}^{4} H_i^- \cup \Omega^o) \cap Q| \leq \sum_{i=1}^{4} |H_1^- \cap Q| + |\Omega^o \cap Q| \\ \leq& \sum_{i=1}^{4} |H_1^- \cap Q| + |\widetilde{\Omega} \cap Q| \leq \frac{4n}{24} + 0.67n < n, \end{aligned} \tag{2.4}$$

which is a contradiction. Therefore, in $\{x_1 - x_2 = 0, x_1 - x_3 = 0, x_2 = 0, x_3 = 0\}$ there exists at lease one plane such that any closed half space with this line as the boundary contains at least $\frac{n}{24}$ points of Q.

The second step is to obtain three sets of planes which have the same structure with $\mathcal{H}_3$, and this can be done through orthogonal transformation. Since a plane through the origin can be uniquely determined by its normal vector, we can use normal vectors $V_3 = \{(1,0,0),(0,1,0),(0,0,1),(1,\pm1,0),(0,1,\pm1),(\pm1,0,1)\}$ to represent planes in $\mathcal{H}_3$. Then, we choose three orthogonal matrices M_1, M_2, M_3 and define $V_3(M_i) = \{vM_i |\ v \in V_3\}$ for $i = 1,2,3$. One set of feasible orthogonal matrices is $\{M_i |\ M_i = I_3 - 2u_{3,i}u_{3,i}^T$, for $i = 1,2,3\}$, where I_3 is a 3×3 identity matrix, and $u_{3,i} = (1^i, 2^i, 3^i)^T$ is a column vector. It can be verified that $\min_{v_i \in V_3(M_i), \forall\ i \in [3]} |\mathbf{Det}([v_1; v_2; v_3])| \geq 4.8468 \times 10^{-4}$, where $[v_1; v_2; v_3]$ is a 3×3 matrix and v_i is its ith row. This means if we arbitrarily choose three vectors v_1, v_2, v_3 from $V_3(M_1)$, $V_3(M_2)$ and $V_3(M_3)$ respectively, then these three vectors are linearly independent, so the three planes determined by these vectors can form an oblique coordinates system. We can use the method in the proof of Theorem 2.6, to generate a set $T(v_1, v_2, v_3)$ with size $O(C_{[v_1;v_2;v_3]} \frac{k^3}{\varepsilon^3})$ in this oblique coordinate system, where $C_{[v_1;v_2;v_3]}$ is a constant determined by $|\mathbf{Det}([v1; v2; v3])|$. For the three orthogonal matrices we chose above, $|\mathbf{Det}([v1; v2; v3])|$ has a lower bound, so the constant $C_{[v_1;v_2;v_3]}$ has an upper bound, which implies $O\left(C_{[v_1;v_2;v_3]} \frac{k^3}{\varepsilon^3}\right) = O\left(\frac{k^3}{\varepsilon^3}\right)$.

For any L_1 median p of n points Q and any $V_3(M_i)$ there must be a plane through p and orthogonal to a vector in $V_3(M_i)$ such that in both sides of this plane there are at least $\frac{n}{24}$ points of Q. So, there exist $(v_1, v_2, v_3) \in V_3(M_1) \times V_3(M_2) \times V_3(M_3)$ and $x \in T(v_1, v_2, v_3)$ such that $\|x - p\| \leq \varepsilon \text{cost}(p, Q)$. Therefore, we can take $T = \cup_{v_i \in V_3(M_i), \forall\ i \in [3]} T(v_1, v_3, v_3)$ as an ε-support for L_1 median on $\mathcal{P}$ with size $O(\frac{k^3}{\varepsilon^3})$. □

In the proof of Theorem 2.7, we choose three orthogonal matrices M_1, M_2, M_3. These

three matrices are independent from the input data $\mathcal{P}$, so we can store these orthogonal matrices and use them to compute the ε-support of L_1 median for any $\mathcal{P}$ in $\mathbb{R}^3$.

To generalize the result of Theorem 2.7 to $\mathbb{R}^d$, we can use $d + 2\binom{n}{2}$ hyperplanes $\mathcal{H}_d = \{x_i = 0 \mid i \in [d]\} \cup \{x_i \pm x_j = 0 \mid 1 \leq i < j \leq d\}$ to partition $\mathbb{R}^d$ into $d2^d$ regions: $\{\Omega_{i,\mathbf{s}} \mid i \in [d], \mathbf{s} \in \{1, -1\}^d\}$, where $\Omega_{i,\mathbf{s}} = \Omega_{i,(s_1,\cdots,s_d)} := \{(x_1, \cdots, x_d) \in \mathbb{R}^d \mid s_i x_i \geq s_j x_j \geq 0, \forall\, j \in [d]\}$. All of these regions have the same shape with $\Omega_{1,(1,\cdots,1)} = \{(x_1, \cdots, x_d) \in \mathbb{R}^d \mid x_1 \geq x_j \geq 0, \text{for } j = 2, \cdots, d\}$. Using the method in the proof of Theorem 2.7 we can show, if p is the L_1 median of n points Q and is the origin, then there is a hyperplane h in $\mathcal{H}_d$ such that any half space with h as the boundary contains at least $\frac{n}{d2^d}$ points of Q. (In $\mathbb{R}^d$, (2.4) will become $n \leq \frac{2(d-1)n}{d2^d} + \frac{\sqrt{d}n}{1+\sqrt{d}} \frac{d2^d+1}{d2^d}$, and it is easy to show the right side of this inequality is always less than n for all $d \geq 3$, so the method in the proof of Theorem 2.7 still works.)

Suppose V_d is the collection of normal vectors of all hyperplanes in $\mathcal{H}_d$. We randomly choose a set of d-dimensional orthogonal matrices $\mathcal{M} = \{M_1, \cdots, M_d\}$, and define $V_d(M_i) = \{vM_i \mid v \in V_d\}$ for $i = 1, \cdots, d$. If $\min_{v_i \in V_d(M_i), \forall\, i \in [d]} \mid \mathbf{Det}([v_1; \cdots; v_d])| \geq c_{\mathcal{M}} > 0$, where $c_{\mathcal{M}}$ is a positive constant dependent on $\mathcal{M}$ and $[v_1; \cdots; v_d]$ is a $d \times d$ matrix with v_i as its ith row, then we can store these matrices, for each $(v_1, \cdots, v_d) \in V_d(M_1) \times \cdots \times V_d(M_d)$ build an oblique coordinate system, and use the method in Theorem 2.6, to generate a set $T(v_1, \cdots, v_d)$ with size $O(C_{[v_1;\cdots;v_d]} \frac{k^d}{\varepsilon^d}) = O(C_{\mathcal{M}} \frac{k^d}{\varepsilon^d})$, where $C_{\mathcal{M}}$ is a positive constant dependent on $\mathcal{M}$. Finally, we return $T = \cup_{v_i \in V_d(M_i), \forall\, i \in [d]} T(v_1, \cdots, v_d)$ as an ε-support for L_1 median on $\mathcal{P}$, and the size of T is $|T| = O(C_{\mathcal{M}} \frac{k^d}{\varepsilon^d}) = O(\frac{k^d}{\varepsilon^d})$, since $\mathcal{M}$ is fixed for all uncertain data in $\mathbb{R}^d$.

Since a d-dimensional orthogonal matrix has $d(d-1)/2$ independent variables, we can always find orthogonal matrices $M_1, \cdots, M_d$ and a constant $c_{\mathcal{M}}$, such that

$$\min_{v_i \in V_d(M_i), \forall\, i \in [d]} |\mathbf{Det}([v_1; \cdots; v_d])| \geq c_{\mathcal{M}} > 0,$$

and for fixed d, $M_1, \cdots, M_d$ can be stored to deal with any input data $\mathcal{P}$ in $\mathbb{R}^d$. For example, for $d = 4$ we can define $M_i = I_4 - 2u_{4,i}u_{4,i}^T$, for $i = 1, \cdots, 4$, where I_4 is an identity matrix and $u_{4,i} = (1^i, 2^i, 3^i, 4^i)^T$, and it can be verified that $\min_{v_i \in V_4(M_i), \forall\, i \in [4]} |\mathbf{Det}([v_1; \cdots; v_4])| \geq 3.7649 \times 10^{-6}$.

For $d = 5$, we can define $M_i = I_5 - 2u_{5,i}u_{5,i}^T$, for $i = 1, \cdots, 5$, where I_5 is an identity

matrix and $u_{5,i} = (1^i, 2^i, 3^i, 4^i, 5^i)^T$, and we have $\min_{v_i \in V_5(M_i), \forall\ i \in [5]} |\mathbf{Det}([v_1; \cdots; v_5])| \geq 2.3635 \times 10^{-11}$. In summary, we have the following theorem.

Theorem 2.8. *Given a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^d$, and $\varepsilon \in (0, 1]$, for any fixed d we can construct an ε-support T for L_1 median on $\mathcal{P}$ that has a size $|T| = O\left(\frac{k^d}{\varepsilon^d}\right)$.*

2.3.6 Assigning a Weight to T in $\mathbb{R}^1$

Here we provide an algorithm to assign a weight to T in $\mathbb{R}^1$, which approximates the probability distribution of median. For T in $\mathbb{R}^d$, we provide a randomized algorithm in Section 2.4.1.

Define the weight of $p_{i,j} \in P_{\mathsf{flat}}$ as $w(p_{i,j}) = \frac{1}{k^n}|\{Q \Subset \mathcal{P} \mid p_{i,j} \text{ is the median of Q}\}|$, the probability it is the median. Suppose T is constructed by our greedy algorithm for $\mathbb{R}^1$. For $p_{i,j} \in P_{\mathsf{flat}}$, we introduce a map $f_T : P_{\mathsf{flat}} \to T$,

$$f_T(p_{i,j}) = \arg\min\{|x - p_{i,j}| \mid x \in T,\ |x - p_{i,j}| \leq \varepsilon \mathsf{cost}(p_{i,j})\},$$

where $\mathsf{cost}(p_{i,j}) = \min\{\mathsf{cost}(p_{i,j}, Q) \mid p_{i,j} \text{ is the median of } Q \text{ and } Q \Subset \mathcal{P}\}$.

Intuitively, this maps each $p_{i,j} \in P_{\mathsf{flat}}$ onto the closest point $x \in T$, unless it violates the ε-approximation property which another further point satisfies.

Now for each $x \in T$, define weight of x as $\hat{w}(x) = \sum_{\{p_{i,j} \in P_{\mathsf{flat}} | f_T(p_{i,j}) = x\}} w(p_{i,j})$. So we first compute the weight of each point in P_{flat} and then obtain the weight of points in T in another linear sweep. Our ability to calculate the weights w for each point in P_{flat} is summarized in the next lemma. The algorithm, explained within the proof, is a dynamic program that expands a specific polynomial similar to Li *et.al.* [64], where in the final state, the coefficients correspond with the probability of each point being the median.

Lemma 2.2. *We can output $w(p_{i,j})$ for all points in P_{flat} in $\mathbb{R}^1$ in $O(n^2k)$ time.*

Proof. For any $p_{i_0} \in P_{i_0}$, we define the following terms to count the number of points to the left (l_j) or right (r_j) of it in the jth uncertain point (excluding P_{i_0}):

$$l_j = \begin{cases} |\{p \in P_j \mid p \leq p_{i_0}\}| & \text{if } 1 \leq j \leq i_0 - 1 \\ |\{p \in P_{j+1} \mid p \leq p_{i_0}\}| & \text{if } i_0 \leq j \leq n-1 \end{cases},$$

$$r_j = \begin{cases} |\{p \in P_j \mid p \geq p_{i_0}\}| & \text{if } 1 \leq j \leq i_0 - 1 \\ |\{p \in P_{j+1} \mid p \geq p_{i_0}\}| & \text{if } i_0 \leq j \leq n-1 \end{cases}$$

. Then, if n is odd, we can write the weight of p_{i_0} as

$$w(p_{i_0}) = \frac{1}{k^n} \sum_{\substack{S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{1, \cdots, n-1\}}} (l_{i_1} \cdot l_{i_2} \cdot \ldots \cdot l_{i_{\frac{n-1}{2}}} \cdot r_{j_1} \cdot r_{j_2} \cdot \ldots \cdot r_{j_{\frac{n-1}{2}}}),$$

where $S_1 = \{i_1, i_2, \cdots, i_{\frac{n-1}{2}}\}$ and $S_2 = \{j_1, j_2, \cdots, j_{\frac{n-1}{2}}\}$. This sums over all partitions S_1, S_2 of uncertain points on the left or right of p_{i_0} for which it is the median, and each term is the product of ways each uncertain point can be on the appropriate side. We define $w(p_{i_0})$ similarly when n is even, then the last index of S_2 is $j_{\frac{n}{2}}$.

We next describe the algorithm for n odd; the case for n even is similar. To compute $\sum_{\substack{S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{1, \cdots, n-1\}}} (l_{i_1} \cdot l_{i_2} \cdot \ldots \cdot l_{i_{\frac{n-1}{2}}} \cdot r_{j_1} \cdot r_{j_2} \cdot \ldots \cdot r_{j_{\frac{n-1}{2}}})$, we consider the following polynomial:

$$(l_1 x + r_1)(l_2 x + r_2) \cdots (l_{n-1} x + r_{n-1}), \tag{2.5}$$

where $\sum_{\substack{S_1 \cap S_2 = \emptyset \\ S_1 \cup S_2 = \{1, \cdots, n-1\}}} (l_{i_1} \cdot l_{i_2} \cdot \ldots \cdot l_{i_{\frac{n-1}{2}}} \cdot r_{j_1} \cdot r_{j_2} \cdot \ldots \cdot r_{j_{\frac{n-1}{2}}})$ is the coefficient of $x^{\frac{n-1}{2}}$. We define $\rho_{i,j}$ $(1 \leq i \leq n-1, 0 \leq j \leq i)$ as the coefficient of x^j in the polynomial $(l_1 x + r_1) \cdots (l_i x + r_i)$ and then it is easy to check $\rho_{i,j} = l_i \rho_{i-1,j-1} + r_i \rho_{i-1,j}$. Thus we can use dynamic programming to compute $\rho_{n-1,0}, \rho_{n-1,1}, \cdots, \rho_{n-1,n-1}$, as shown in Algorithm 2.1.

Algorithm 2.1 Compute $\rho_{n-1,0}, \rho_{n-1,1}, \cdots, \rho_{n-1,n-1}$

```
Let ρ_{1,0} = r_1, ρ_{1,1} = l_1, ρ_{1,2} = 0.
for i = 2 to n − 1 do
  for j = 0 to i do
    ρ_{i,j} = l_i ρ_{i−1,j−1} + r_i ρ_{i−1,j}
  ρ_{i,i+1} = 0
return ρ_{n−1,0}, ρ_{n−1,1}, ··· , ρ_{n−1,n−1}.
```

Thus Algorithm 2.1 computes the weight $\frac{1}{k^n} w(p_{i_0}) = \rho_{n-1,\frac{n-1}{2}}$ for a single $p_{i_0} \in P_{\mathsf{flat}}$. Next we show, we can reuse much of the structure to compute the weight for another point; this will ultimately shave a factor n off of running Algorithm 2.1 nk times.

Suppose for $p_{i_0} \in P_{i_0}$ we have obtained $\rho_{n-1,0}, \rho_{n-1,1}, \ldots, \rho_{n-1,n-1}$ by Algorithm 2.1, and then we consider $p_{i'_0} = \min\{p \in P_{\mathsf{flat}} \setminus P_{i_0} \mid p \geq p_{i_0}\}$. We assume $p_{i'_0} \in P_{i'_0}$, and if $i'_0 < i_0$, we construct a polynomial

$$(l_1 x + r_1) \cdots (l_{i'_0-1} x + r_{i'_0-1})(\tilde{l}_{i'_0} x + \tilde{r}_{i'_0})(l_{i'_0+1} x + r_{i'_0+1}) \cdots (l_{n-1} x + r_{n-1}) \tag{2.6}$$

and if $i'_0 > i_0$, we construct a polynomial

$$(l_1 x + r_1) \cdots (l_{i'_0-2} x + r_{i'_0-2})(\tilde{l}_{i'_0-1} x + \tilde{r}_{i'_0-1})(l_{i'_0} x + r_{i'_0}) \cdots (l_{n-1} x + r_{n-1}) \tag{2.7}$$

where $\tilde{l}_{i'_0} = \tilde{l}_{i'_0-1} = |\{p \in P_{i_0} \mid p \leq p_{i'_0}\}|$ and $\tilde{r}_{i'_0} = \tilde{r}_{i'_0-1} = |\{p \in P_{i_0} \mid p \geq p_{i'_0}\}|$.

Since (2.5) and (2.6) have only one different factor, we obtain the coefficients of (2.6) from the coefficients of (2.5) in $O(n)$ time. We recover the coefficients of $(l_1 x + r_1) \cdots (l_{i'-1} x + r_{i'-1})(l_{i'_0+1} x + r_{i'_0+1}) \cdots (l_{n-1} x + r_{n-1})$ from $\rho_{n-1,0}, \rho_{n-1,1}, \cdots, \rho_{n-1,n-1}$, and then use these coefficients to compute the coefficients of (2.6). Similarly, if $i'_0 > i_0$, we obtain the coefficients of (2.7) from the coefficients of (2.5). Therefore, we can use $O(n^2)$ time to compute the weight of the first point in P_{flat} and then use $O(n)$ time to compute the weight of each other point. The whole time is $O(n^2) + nkO(n) = O(n^2 k)$. □

Corollary 2.2. *We can assign $\hat{w}(x)$ to each $x \in T$ in $\mathbb{R}^1$ in $O(n^2 k)$ time.*

2.4 A Randomized Algorithm to Construct a Covering Set

In this section we describe a much more general randomized algorithm for robust estimators on uncertain data. It constructs an approximate covering set of the support of the distribution of the estimator, and estimates the weight at the same time. The support of the distribution is not as precise compared to the techniques in the previous section in that the new technique may fail to cover regions with small probability of containing the estimator.

Suppose $\mathcal{P} = \{P_1, P_2, \cdots, P_n\}$ is a set of uncertain data, where for $i \in [n]$, $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\} \subseteq \mathcal{X}$ for some domain $\mathcal{X}$. An estimator $E: \{Q \mid Q \Subset \mathcal{P}\} \mapsto Y$ maps $Q \Subset \mathcal{P}$ to a metric space (Y, φ). Let $B(y, r) = \{y' \in Y \mid \varphi(y, y') \leq r\}$ be a ball of radius r in that metric space. We denote ν as the VC-dimension of the range space $(Y, \mathcal{R})$ induced by these balls, with $\mathcal{R} = \{B(y, r) \mid y \in Y, r \geq 0\}$.

We now analyze the simple algorithm which randomly instantiates traversals $Q \Subset \mathcal{P}$, and constructors their estimators $z = E(Q)$. Repeating this N times builds a domain

$T = \{z_1, z_2, \ldots, z_N\}$ each with weight $w(z_i) = 1/N$. Duplicates of domain points can have their weights merged as described in Algorithm 2.2.

Algorithm 2.2 Approximate the weight of points in T

```
Initialize T = ∅
for j = 1 to N do
    Randomly choose Q ⋐ 𝒫, and set z = E(Q).
    if z = z′ for some z′ ∈ T, then increment c_z′ = c_z′ + 1
    else add z to T, and set c_z = 1.
return c_z/N as the approximate value of w(z) for all z ∈ T
```

Theorem 2.9. *For $\varepsilon > 0$ and $\delta \in (0,1)$, set $N = O((1/\varepsilon^2)(\nu + \log(1/\delta)))$. Then, with probability at least $1-\delta$, for any $B \in \mathcal{R}$ we have $|\sum_{z \in T \cap B} w(z) - \Pr_{Q \Subset \mathcal{P}}[E(Q) \in B]| \leq \varepsilon$.*

Proof. Let T^* be the true support of $E(Q)$ where $Q \Subset P$, and let $w^* : T^* \to \mathbb{R}^+$ be the true probability distribution defined on T^*; e.g., for discrete T^*, then for any $z' \in T^*$, $w^*(z') = \Pr_{Q \Subset P}[E(Q) = z']$. Then each random z generated is a random draw from w^*. Hence for a range space with bounded VC-dimension [85] ν, we can apply the sampling bound [65] for ε-approximations of these range spaces to prove our claim. □

In Theorem 2.9, for $z_i \in T$, if we choose $B = B(z_i, r) \in \mathcal{R}$ with r small enough such that $T \cap B$ only contains z_i, then we obtain the following.

Corollary 2.3. *For $\varepsilon > 0$ and $\delta \in (0,1)$, set $N = O((1/\varepsilon^2)(\nu + \log(1/\delta)))$. Then, with probability at least $1-\delta$, for any $z \in Y$ we have $|w(z) - \Pr_{Q \Subset \mathcal{P}}[E(Q) = z]| \leq \varepsilon$.*

Remark 2.3. *We can typically define a metric space (Y, φ) where $\nu = O(1)$; for instance for point estimators (e.g., the geometric median), define a projection into $\mathbb{R}^1$ so no z_is map to the same point, then define distance φ as restricted to the distance along this line, so metric balls are intervals (or slabs in $\mathbb{R}^d$); these have $\nu = 2$.*

2.4.1 Application to Geometric Median

For each $Q \Subset \mathcal{P}$, the geometric median m_Q may take a distinct value. Thus even calculating that set, let alone their weights in the case of duplicates, would require at least $\Omega(k^n)$ time. But it is straightforward to apply this randomized approach. For $P_{\mathsf{flat}} \in \mathbb{R}^d$, the natural metric space (Y, φ) is $Y = \mathbb{R}^d$ and φ as the Euclidian distance.

However, there is no known closed form solution for the geometric median; it can be computed within any additive error ϕ through various methods [12, 14, 17, 88]. As such, we can state a slightly more intricate corollary.

Corollary 2.4. *Set $\varepsilon > 0$ and $\delta \in (0,1)$ and $N = O((1/\varepsilon^2)(d + \log(1/\delta)))$. For an uncertain point set $\mathcal{P}$ with $P_{\mathsf{flat}} \subset \mathbb{R}^d$, let the estimator E be the geometric median, and let E_ϕ be an algorithm that finds an approximation to the geometric median within additive error $\phi > 0$. Run the algorithm using E_ϕ. Then for any ball $B = B(x,r) \in \mathcal{R}$, there exists[2] another ball $B' = B(x,r')$ with $|r - r'| \leq \phi$ such that with probability at least $1 - \delta$,*

$$\Big| \sum_{z \in T \cap B'} w(z) - \Pr_{Q \Subset \mathcal{P}}[E(Q) \in B] \Big| \leq \varepsilon.$$

2.4.2 Application to Siegel Estimator

The Siegel (repeated median) estimator [81] is a robust estimator S for linear regression in $\mathbb{R}^2$ with optimal breakdown point 0.5. For a set of points Q, for each $q_i \in Q$ it computes slopes of all lines through q_i and each other $q' \in Q$, and takes their median a_i. Then it takes the median a of the set $\{a_i\}_i$ of all median slopes. The offset b of the estimated line $\ell : y = ax + b$, is the median of $(y_i - ax_i)$ for all points $q_i = (x_i, y_i)$. For uncertain data $P_{\mathsf{flat}} \subset \mathbb{R}^2$, we can directly apply our general technique for this estimator.

We use the following metric space (Y, φ). Let $Y = \{\ell \mid \ell \text{ is a line in } \mathbb{R}^2 \text{ with form } y = ax + b, \text{ where } a, b \in \mathbb{R}\}$. Then let φ be the Euclidean distance in the standard dual; for two lines $\ell : y = ax + b$ and $\ell' : y = a'x + b'$, define $\varphi(\ell, \ell') = \sqrt{(a - a')^2 + (b - b')^2}$. By examining the dual space, we see that $(Y, \mathcal{R})$ with $\mathcal{R} = \{B(\ell, r) \mid \ell \in Y, r \geq 0\}$ and $B(\ell, r) = \{\ell' \in Y \mid \varphi(\ell, \ell') \leq r\}$ has a VC-dimension 3.

From the definition of the Siegel estimator [81], there can be at most $O(n^3k^3)$ distinct lines in $T = \{S(Q) \mid Q \Subset \mathcal{P}\}$. By Corollary 2.3, setting $N = O((1/\varepsilon^2)\log(1/\delta))$, then with probability at least $1 - \delta$ for all $z \in T$ we have $\Big| w(z) - \Pr_{Q \Subset \mathcal{P}}[S(Q) = z] \Big| \leq \varepsilon$.

[2]To simplify the discussion on degenerate behavior, define ball B', so any point q on its boundary can be defined inside or outside of B, and this decision can be different for each q, even if they are co-located.

CHAPTER 3

THE ROBUSTNESS OF ESTIMATOR COMPOSITION

3.1 Introduction

Robust statistical estimators [51, 56] (in particular, resistant estimators), such as the median, are an essential tool in data analysis since they are provably immune to outliers. Given data with a large fraction of extreme outliers, a robust estimator guarantees the returned value is still within the non-outlier part of the data. In particular, the role of these estimators is quickly growing in importance as the scale and automation associated with data collection and data processing becomes more commonplace. Artisanal data (hand crafted and carefully curated), where potential outliers can be removed, is becoming proportionally less common. Instead, important decisions are being made blindly based on the output of analysis functions, often without looking at individual data points and their effect on the outcome. Thus using estimators as part of this pipeline that are not robust are susceptible to erroneous and dangerous decisions as the result of a few extreme and rogue data points.

Although other approaches like regularization and pruning a constant number of obvious outliers are common as well, they do not come with the important guarantees that ensure these unwanted outcomes absolutely cannot occur.

In this chapter, we initiate the formal study of the robustness of composition of estimators through the notion of breakdown points. These are especially important with the growth of data analysis pipelines where the final result or prediction is the result of several layers of data processing. When each layer in this pipeline is modeled as an estimator, then our analysis provides the first general robustness analysis of these processes.

The *breakdown point* [32, 52] is a basic measure of robustness of an estimator. Intuitively, it

describes how many outliers can be in the data without the estimator becoming unreliable. However, the literature is full of slightly inconsistent and informal definitions of this concept. For example:

- Aloupis [7] write "the breakdown point is the proportion of data which must be moved to infinity so that the estimator will do the same."
- Huber and Ronchetti [57] write "the breakdown point is the smallest fraction of bad observations that may cause an estimator to take on arbitrarily large aberrant values."
- Dasgupta, Kumar, and Srikumar [90] write "the breakdown point of an estimator is the largest fraction of the data that can be moved arbitrarily without perturbing the estimator to the boundary of the parameter space."

All of these definitions have similar meanings, and they are typically sufficient for the purpose of understanding a single estimator. However, they are not mathematically rigorous, and it is difficult to use them to discuss the breakdown point of composite estimators.

Composition of Estimators. In a bit more detail (we give formal definitions in Section 3.2.1), an estimator E maps a data set to single value in another space, sometimes the same as a single data point. For instance the mean or the median are simple estimators on one-dimensional data. A composite E_1-E_2 estimator applies two estimators E_1 and E_2 on data stored in a hierarchy. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_n\}$ be a set of subdata sets, where each subdata set $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k}\}$ has individual data readings. Then the E_1-E_2 estimator reports $E_2(E_1(P_1), E_1(P_2), \ldots, E_1(P_n))$, that is the estimator E_2 applied to the output of estimator E_1 on each subdata set.

3.1.1 Examples of Estimator Composition

Composite estimators arise in many scenarios in data analysis.

Uncertain Data. For instance, in the last decade there has been increased focus on the study of uncertainty data [28, 60, 79] where instead of analyzing a data set, we are given a model of the uncertainty of each data point. Consider tracking the summarization of a group of n people based on noisy GPS measurements. For each person i we might get k

readings of their location P_i, and use these k readings as a discrete probability distribution of where that person might be. Then in order to represent the center of this set of people a natural thing to do would be to estimate the location of each person as $x_i \leftarrow E_1(P_i)$, and then use these estimates to summarize the entire group $E_2(x_1, x_2, \ldots, x_n)$. Using the mean as E_1 and E_2 would be easy, but would be susceptible to even a single outrageous outlier (all people are in Manhattan, but a spurious reading was at $(0, 0)$ lat-long, off the coast of Africa). An alternative is to use the L_1-median for E_1 and E_2, that is known to have an optimal breakdown point of 0.5. But what is the breakdown point of the E_1-E_2 estimator?

Robust Analysis of Bursty Behavior. Understanding the robustness of estimators can also be critical towards how much one can "game" a system. For instance, consider a start-up media website that gets bursts of traffic from memes they curate. They publish a statistic showing the median of the top half of traffic days each month, and aggregate these by taking the median of such values over the top half of all months. This is a composite estimator, and they proudly claim, even through they have bursty traffic, it is robust (each estimator has a breakdown point of 0.25). If this composite estimator shows large traffic, should a potential buyer of this website be impressed? Is there a better, more robust estimator the potential buyer could request? If the media website can stagger the release of its content, how should they distribute it to maximize this composite estimator?

Part of the Data Analysis Pipeline. This process of estimator composition is very common in broad data analysis literature. This arises from the idea of an "analysis pipeline" where at several stages estimators or analysis is performed on data, and then further estimators and analysis are performed downstream. In many cases a robust estimator like the median is used, specifically for its robustness properties, but there is no analysis of how robust the composition of these estimators is.

3.1.2 Main Results

This chapter initiates the formal and general study of the robustness of composite estimators.

- In Subsection 3.2.1, we give two formal definitions of breakdown points which are both required to prove composition theorem. One variant of the definition closely

aligns with other formalizations [32, 52], while another is fundamentally different.

- The main result provides general conditions under which an E_1-E_2 estimator with breakdown points β_1 and β_2, has a breakdown point of $\beta_1\beta_2$ (Theorem 3.2 in Subsection 3.2.2).

- Moreover, by showing examples where our conditions do not strictly apply, we gain an understanding of how to circumvent the above result. An example is in composite percentile estimators (e.g., E_1 returns the 25th percentile, and E_2 the 75th percentile of a ranked set). These composite estimators have larger breakdown point than $\beta_1 \cdot \beta_2$.

- The main result can extended to multiple compositions, under suitable conditions, so for instance an E_1-E_2-E_3 estimator has a breakdown point of $\beta_1\beta_2\beta_3$ (Theorem 3.3 in Subsection 3.2.3). This implies that long analysis chains can be very suspect to a few carefully places outliers since the breakdown point decays exponentially in the length of the analysis chain.

- In Section 3.3, we highlight several applications of this theory, including robust regression, robustness of p-values, a depth-3 composition, and how to advantageously manipulate the observation about percentile estimator composition. We demonstrate a few more applications with simulations in Section 3.4.

3.2 Robustness of Estimator Composition

3.2.1 Formal Definitions of Breakdown Points

In this chapter, we give two definitions for the breakdown point: *Asymptotic Breakdown Point* and *Asymptotic Onto-Breakdown Point*. The first definition, Asymptotic Breakdown Point, is similar to the classic formal definitions in [52] and [32] (including their highly technical nature), although their definitions of the estimator are slightly different leading to some minor differences in special cases. However our second definition, Asymptotic Onto-Breakdown Point, is a structurally new definition, and we illustrate how it can result in significantly different values on some common and useful estimators. Our main theorem will require both definitions, and the differences in performance will lead to several new applications and insights.

We define an *estimator* E as a function from the collection of some finite subsets of a metric space $(\mathscr{X}, d)$ to another metric space $(\mathscr{X}', d')$:

$$E: \mathscr{A} \subset \{X \subset \mathscr{X} \mid 0 < |X| < \infty\} \mapsto \mathscr{X}', \tag{3.1}$$

where X is a multiset. This means if $x \in X$ then x can appear more than once in X, and the multiplicity of elements will be considered when we compute $|X|$.

Finite Sample Breakdown Point. For estimator E defined in (3.1) and positive integer n we define its *finite sample breakdown point* $g_E(n)$ over a set M as

$$g_E(n) = \begin{cases} \max(M) & \text{if } M \neq \emptyset \\ 0 & \text{if } M = \emptyset \end{cases} \tag{3.2}$$

where for $\rho(x', X) = \max_{x \in X} d(x', x)$ is the distance from x' to the furthest point in X,

$$\begin{aligned} M = \{m \in [0, n] \mid \forall X \in \mathscr{A}, |X| = n, \forall\, G_1 > 0, \exists\, G_2 = G_2(X, G_1) \text{ s.t.} \\ \forall X' \in \mathscr{A}, \text{ if } |X'| = n \text{ and } |\{x' \in X' \mid \rho(x', X) > G_1\}| \leq m \\ \text{then } d'(E(X), E(X')) \leq G_2\}. \end{aligned} \tag{3.3}$$

For an estimator E in (3.1) and $X \in \mathscr{A}$, the finite sample breakdown point $g_E(n)$ means if the number of unbounded points in X' is at most $g_E(n)$, then $E(X')$ will be bounded. Lets break this definition down a bit more. The definition holds over all data sets $X \in \mathscr{A}$ of size n, and for all values $G_1 > 0$ and some value G_2 defined as a function $G_2(X, G_1)$ of the data set X and value G_1. Then $g_E(n)$ is the maximum value m (over all X, G_1, and G_2 above) such that for all $X' \in \mathscr{A}$ with $|X'| = n$ then $|\{x' \in X' \mid \rho(x', X) > G_1\}| \leq m$ (that is at most m points are further than G_1 from X) where the estimators are close, $d'(E(X), E(X')) \leq G_2$.

For example, consider a point set $X = \{0, 0.15, 0.2, 0.25, 0.4, 0.55, 0.6, 0.65, 0.72, 0.8, 1.0\}$ with $n = 11$ and median 0.55. If we set $G_1 = 3$, then we can consider sets X' of size 11 with fewer than m points that are either greater than 3 or less than -2. This means in X' there are at most m points which are greater than 3 or less than -2, and all other $n - m$ points are in $[-2, 3]$. Under these conditions, we can (conservatively) set $G_2 = 4$, and know that for values of m as $1, 2, 3, 4$, or 5, then the median of X' must be between -3.45 and 4.55; and this holds no matter where we set those m points (e.g., at 20 or at 1000). This does not hold for $m \geq 6$, so $g_E(11) = 5$.

Asymptotic Breakdown Point. If the limit $\lim_{n\to\infty} \frac{g_E(n)}{n}$ exists, then we define this limit

$$\beta = \lim_{n\to\infty} \frac{g_E(n)}{n} \tag{3.4}$$

as the *asymptotic breakdown point*, or *breakdown point* for short, of the estimator E.

Remark 3.1. *It is not hard to see that many common estimators satisfy the conditions. For example, the median, L_1-median [7], and Siegel estimators [81] all have asymptotic breakdown points of* 0.5.

Asymptotic Onto-Breakdown Point. For an estimator E given in (3.1) and positive integer n, if

$$\begin{aligned}\widetilde{M} = \{0 \le m \le n \mid \forall\, X \in \mathscr{A}, |X| = n, \forall\, y \in \mathscr{X}',\\ \exists\, X' \in \mathscr{A} \text{ s.t. } |X'| = n, |X \cap X'| = n - m, E(X') = y\}\end{aligned}$$

is not empty, we define

$$f_E(n) = \min(\widetilde{M}). \tag{3.5}$$

The definition of $f_E(n)$ implies, if we change $f_E(n)$ elements in X, we can make E become *any* value in $\mathscr{X}'$: it is onto. In contrast $g_E(n)$ only requires $E(X')$ to become far from $E(X)$, perhaps only in one direction. Then the *asymptotic onto-breakdown point* is defined as the following limit if it exists

$$\lim_{n\to\infty} \frac{f_E(n)}{n}. \tag{3.6}$$

Remark 3.2. *For a quantile estimator E that returns a percentile other than the 50th, then $\lim_{n\to\infty} \frac{g_E(n)}{n} \neq \lim_{n\to\infty} \frac{f_E(n)}{n}$. For instance, if E returns the 25th percentile of a ranked set, setting only 25% of the data points to $-\infty$ causes E to return $-\infty$; hence $\lim_{n\to\infty} \frac{g_E(n)}{n} = 0.25$. And while any value less than the original 25th percentile can also be obtained; to return a value larger than the largest element in the original set, at least 75% of the data must be modified, thus $\lim_{n\to\infty} \frac{f_E(n)}{n} = 0.75$.*

As we will observe in Section 3.3, this nuance in definition regarding percentile estimators will allow for some interesting composite estimator design.

3.2.2 Definition of $E1$-$E2$ Estimators, and their Robustness

We consider the following two estimators:

$$E_1 : \mathscr{A}_1 \subset \{X \subset \mathscr{X}_1 \mid 0 < |X| < \infty\} \mapsto \mathscr{X}_2, \tag{3.7}$$

$$E_2 : \mathscr{A}_2 \subset \{X \subset \mathscr{X}_2 \mid 0 < |X| < \infty\} \mapsto \mathscr{X}_2', \tag{3.8}$$

where any finite subset of $E_1(\mathscr{A}_1)$, the range of E_1, belongs to $\mathscr{A}_2$. Suppose $P_i \in \mathscr{A}_1$, $|P_i| = k$ for $i = 1, 2, \cdots, n$ and $P_{\mathsf{flat}} = \uplus_{i=1}^n P_i$, where $\uplus$ means if x appears n_1 times in X_1 and n_2 times in X_2 then x appears $n_1 + n_2$ times in $X_1 \uplus X_2$. We define

$$E(P_{\mathsf{flat}}) = E_2\left(E_1(P_1), E_1(P_2), \cdots, E_1(P_n)\right). \tag{3.9}$$

Theorem 3.1. *Suppose $g_{E_1}(k)$ and $g_{E_2}(n)$ are the finite sample breakdown points of estimators E_1 and E_2 which are given by* (3.7) *and* (3.8) *respectively. If $g_E(nk)$ is the finite sample breakdown point of E given by* (3.9)*, then we have*

$$g_{E_2}(n) g_{E_1}(k) \leq g_E(nk). \tag{3.10}$$

and if

$$\beta_1 = \lim_{k\to\infty} \frac{g_{E_1}(k)}{k}, \;\; \beta_2 = \lim_{n\to\infty} \frac{g_{E_2}(n)}{n}, \beta = \lim_{n,k\to\infty} \frac{g_E(nk)}{nk}$$

and all exist, then

$$\beta_1 \beta_2 \leq \beta. \tag{3.11}$$

Proof. For any fixed $G_1 > 0$, and any subsets $P_1', P_2', \cdots, P_n' \in \mathscr{A}_1$ satisfying $|P_1'| = |P_2'| = \cdots = |P_n'| = k$, and

$$|\{p' \in P_{\mathsf{flat}}' |\, \rho(p', P_{\mathsf{flat}}) > G_1\}| \leq g_{E_2}(n) g_{E_1}(k) \tag{3.12}$$

where $P_{\mathsf{flat}}' = \uplus_{i=1}^n P_i'$, we introduce the notation

$$X = \{E_1(P_1), E_1(P_2), \cdots, E_1(P_n)\}, \quad X' = \{E_1(P_1'), E_1(P_2'), \cdots, E_1(P_n')\}.$$

So, in order to prove (3.10), we only need to bound $E(P_{\mathsf{flat}}')$.

We define

$$I_1 = \left\{1 \leq i \leq n |\, |\{p' \in P_i' |\, \rho(p', P_i) > G_1\}| > g_{E_1}(k)\right\} \tag{3.13}$$

and then have

$$|I_1| \leq g_{E_2}(n). \tag{3.14}$$

Otherwise, since $\rho(p', P_i) > G_1$ implies $\rho(p', P_{\mathsf{flat}}) > G_1$, from $|I_1| > g_{E_2}(n)$ and (3.13) we can obtain

$$|\{p' \in P_{\mathsf{flat}}' |\, \rho(p', P_{\mathsf{flat}}) > G_1\}| > g_{E_2}(n) g_{E_1}(k)$$

which is contradictory to (3.12).

For any $i \notin I_1$, we have $|\{p' \in P_i' |\ \rho(p', P_i) > G_1\}| \leq g_{E_1}(k)$, so, from the definition of $g_{E_1}(k)$ we know

$$\exists\ G_2^i = G_2^i(P_i, G_1),\ \text{s.t. } d_2(E_1(P_i'), E_1(P_i)) \leq G_2^i\ \ \forall\ i \notin I_1.$$

where d_2 is the metric of space $\mathscr{X}_2$. Let

$$G_2 = \max_{i \notin I_1} G_2^i + \max_{1 \leq i,j \leq n} d_2(E_1(P_i), E_1(P_j))$$

then we have

$$\rho(E_1(P_i'), X) \leq G_2, \forall\ i \notin I_1. \tag{3.15}$$

Defining $I_2 = \{1 \leq i \leq n \mid \rho(E_1(P_i'), X) > G_2\}$ from (3.15) we have $I_2 \subset I_1$, which implies $|I_2| \leq |I_1| \leq g_{E_2}(n)$ by (3.14). Therefore, from the definition of $g_{E_2}(n)$, we have

$$\exists\ G_3 = G_3(X, G_2) \text{ s.t. } \|E(P_{\mathsf{flat}}') - E(P_{\mathsf{flat}})\| = \|E_2(X') - E_2(X)\| \leq G_3,$$

which implies (3.10), and (3.11) can be obtained from (3.10) directly. Thus, the proof is completed. □

Remark 3.3. *Under the condition of Theorem 3.1, we cannot guarantee $\beta = \beta_1\beta_2$. For example, suppose E_1 and E_2 take the 25th percentile and the 75th percentile of a ranked set of real numbers respectively. So, we have $\beta_1 = \beta_2 = \frac{1}{4}$. However, $\beta = \frac{1}{4} \cdot \frac{3}{4} = \frac{3}{16}$.*

In fact, the limit of $\frac{g_E(nk)}{nk}$ as $n, k \to \infty$ may even not exist. For example, suppose E_1 takes the 25th percentile of a ranked set of real numbers. When n is odd E_2 takes the the 25th percentile of a ranked set of n real numbers, and when n is even E_2 takes the the 75th percentile of a ranked set of n real numbers. Thus, $\beta_1 = \beta_2 = \frac{1}{4}$, but $g_E(nk) \approx \frac{1}{4}nk$ if n is odd, and $g_E(nk) \approx \frac{1}{4} \cdot \frac{3}{4}nk$ if n is even, which implies $\lim_{n,k\to\infty} \frac{g_E(nk)}{nk}$ does not exist.

Therefore, to guarantee β exist and $\beta = \beta_1\beta_2$, we introduce the definition of asymptotic onto-breakdown point in (3.6). As shown in *Remark 3.2*, the values of (3.4) and (3.6) may be not equal. However, with the condition of the asymptotic breakdown point and asymptotic onto-breakdown point of E_1 being the same, we can finally state our desired clean result.

Theorem 3.2. *For estimators E_1, E_2 and E given by* (3.7), (3.8) *and* (3.9) *respectively, suppose $g_{E_1}(k)$, $g_{E_2}(n)$ and $g_E(nk)$ are defined by* (3.2), *and $f_{E_1}(k)$ is defined by* (3.5). *Moreover, E_1 is an onto function and for any fixed positive integer n we have*

$$\begin{aligned} &\exists\, X \in \mathscr{A}_2, |X| = n, G_1 > 0, s.t.\ \forall\, G_2 > 0, \exists\, X' \in \mathscr{A}_2 \text{ satisfying} \\ &|X'| = n, |X' \setminus X| = g_{E_2}(n) + 1, \text{ and } d_2'(E_2(X), E_2(X')) > G_2. \end{aligned} \tag{3.16}$$

where d_2' is the metric of space $\mathscr{X}_2'$.

If

$$\beta_1 = \lim_{k\to\infty} \frac{g_{E_1}(k)}{k} = \lim_{k\to\infty} \frac{f_{E_1}(k)}{k}, \quad \text{and} \quad \beta_2 = \lim_{n\to\infty} \frac{g_{E_2}(n)}{n} \tag{3.17}$$

both exist, then

$$\beta = \lim_{n,k\to\infty} \frac{g_E(nk)}{nk} \text{ exists} \quad \text{and} \quad \beta = \beta_1\beta_2. \tag{3.18}$$

Proof. For any fixed positive integer n, we can find $X = \{x_1, x_2, \cdots, x_n\} \in \mathscr{A}_2$, and $G_1 > 0$ satisfying (3.16). Since E_1 is an onto function, we can find $P_{\mathsf{flat}} = \uplus_{i=1}^n P_i$ such that $P_i \in \mathscr{A}_1$ and $E_1(P_i) = x_i$ for all $1 \le i \le n$.

From (3.16), we know for any $G_2 > 0$, we can find $X' \in \mathscr{A}_2$ such that $|X'| = n$, $|X' \setminus X| = g_{E_2}(n) + 1$ and

$$d'(E_2(X), E_2(X')) > G_2.$$

This implies the number of different elements between X and X' is $g_{E_2}(n) + 1$. For any $x_i' \in X' \setminus X$, we can find $P_i' \in \mathscr{A}_1$ such that $|P_i'| = k$, $E_1(P_i') = x_i'$ and $|P_i' \setminus P_i| = f_{E_1}(k)$. So, we only need to change $f_{E_1}(k)(g_{E_2}(n) + 1)$ points of P_{flat}, and then we can obtain P_{flat}' such that $|P_{\mathsf{flat}}' \setminus P_{\mathsf{flat}}| = f_{E_1}(k)(g_{E_2}(n) + 1)$ and $d'(E(P_{\mathsf{flat}}), E(P_{\mathsf{flat}}')) > G_2$. This implies

$$g_E(nk) \le f_{E_1}(k)(g_{E_2}(n) + 1). \tag{3.19}$$

Therefore, from Theorem 3.1 and (3.19) we have

$$\frac{g_{E_1}(k)}{k}\frac{g_{E_2}(n)}{n} \le \frac{g_E(nk)}{nk} \le \frac{f_{E_1}(k)}{k}\frac{(g_{E_2}(n) + 1)}{n}. \tag{3.20}$$

Letting n and k go to infinity in (3.20), we obtain (3.18) from (3.17). Thus, the proof of this theorem is completed. □

Remark 3.4. *Without the introduction of $f_E(n)$, we cannot even guarantee $\beta \le \beta_1$ or $\beta \le \beta_2$ only under the condition of Theorem 3.1, even if E_1 and E_2 are both onto functions. For example, for any*

$P = \{p_1, p_2, \cdots, p_k\} \subset \mathbb{R}$ and $X = \{x_1, x_2, \cdots, x_n\} \subset \mathbb{R}$, we define $E_1(P) = 1/median(P)$ (if $median(P) \neq 0$, otherwise define $E_1(P) = 0$) and $E_2(X) = median(y_1, y_2, \cdots, y_n)$, where y_i $(1 \leq y \leq n)$ is given by $y_i = 1/x_i$ (if $x_i \neq 0$, otherwise define $y_i = 0$). Since $g_{E_1}(k) = g_{E_2}(n) = 0$ for all n, k, we have $\beta_1 = \beta_2 = 0$. However, in order to make $E_2(E_1(P_1), E_1(P_2), \cdots, E_1(P_n)) \to +\infty$, we need to make about $\frac{n}{2}$ elements in $\{E(P_1), E(P_2), \cdots, E(P_n)\}$ go to $0+$. To make $E_1(P_i) \to 0+$, we need to make about $\frac{k}{2}$ points in P_i go to $+\infty$. Therefore, we have $g_E(nk) \approx \frac{n}{2} \cdot \frac{k}{2}$ and $\beta = \frac{1}{4}$.

3.2.3 Multi-level Composition of Estimators

To study the breakdown point of composite estimators with more than two levels, we introduce the following estimator:

$$E_3 : \mathscr{A}_3 \subset \{X \subset \mathscr{X}_2' \mid 0 < |X| < \infty\} \mapsto \mathscr{X}_3', \tag{3.21}$$

where any finite subset of $E_2(\mathscr{A}_2)$, the range of E_2, belongs to $\mathscr{A}_3$. Suppose $P_{i,j} \in \mathscr{A}_1$, $|P_{i,j}| = k$ for $i = 1, 2, \cdots, n$, $j = 1, 2, \cdots, m$ and $P^j_{\mathsf{flat}} = \uplus_{i=1}^n P_{i,j}$, $P_{\mathsf{flat}} = \uplus_{j=1}^m P^j_{\mathsf{flat}}$. We define

$$E(P_{\mathsf{flat}}) = E_3\left(E_2(\widetilde{P}^1_{\mathsf{flat}}), E_2(\widetilde{P}^2_{\mathsf{flat}}), \cdots, E_2(\widetilde{P}^m_{\mathsf{flat}})\right), \tag{3.22}$$

where $\widetilde{P}^j_{\mathsf{flat}} = \{E_1(P_{1,j}), E_1(P_{2,j}), \cdots, E_1(P_{n,j})\}$, for $j = 1, 2, \cdots, m$.

From Theorem 3.2, we can obtain the following theorem about the breakdown point of E in (3.22).

Theorem 3.3. *For estimators E_1, E_2, E_3 and E given by* (3.7), (3.8), (3.21) *and* (3.22) *respectively, suppose $g_{E_1}(k)$, $g_{E_2}(n)$, $g_{E_3}(m)$ and $g_E(mnk)$ are defined by* (3.2), *and $f_{E_1}(k)$, $f_{E_2}(n)$ are defined by* (3.5). *Moreover, E_1 and E_2 are both onto functions, and for any fixed positive integer m we have*

$$\exists\, X \in \mathscr{A}_3, |X| = m, G_1 > 0, s.t.\ \forall\, G_2 > 0, \exists\, X' \in \mathscr{A}_3$$
$$\textit{satisfying } |X'| = m, |X' \setminus X| = g_{E_3}(m) + 1, \textit{ and } d_3'(E_3(X), E_3(X')) > G_2.$$

where d_3' is the metric of space $\mathscr{X}_3'$. If

$$\beta_1 = \lim_{k\to\infty} \frac{g_{E_1}(k)}{k} = \lim_{k\to\infty} \frac{f_{E_1}(k)}{k}, \quad \beta_2 = \lim_{n\to\infty} \frac{g_{E_2}(n)}{n} = \lim_{n\to\infty} \frac{f_{E_2}(n)}{n}, \tag{3.23}$$

and $\beta_3 = \lim_{m\to\infty} \frac{g_{E_3}(m)}{m}$ all exist, then

$$\beta = \lim_{m,n,k\to\infty} \frac{g_E(mnk)}{mnk} \textit{ exist} \quad \textit{and} \quad \beta = \beta_1\beta_2\beta_3. \tag{3.24}$$

Proof. We define an estimator $\widetilde{E}$:

$$\widetilde{E}(\widetilde{P}^j_{\mathsf{flat}}) = E_2(E_1(P_{1,j}), E_1(P_{2,j}), \cdots, E_1(P_{n,j}))$$

for $j = 1, 2, \cdots, m$, and first prove the breakdown point of $\widetilde{E}$ is $\tilde{\beta} = \beta_1\beta_2$.

For any fixed $y \in \mathscr{X}_2'$ and $X = \{E_1(P_1), E_1(P_2), \cdots, E_1(P_n)\}$, we can find $X' \in \mathscr{A}_2$ such that $|X'| = n$, $|X \cap X'| = n - f_{E_2}(n)$ and $E_2(X') = y$. For any element $y' \in X' \setminus (X \cap X')$, we can find $E_1(P_i) \in X \setminus (X \cap X')$ and $P_i' \in \mathscr{A}_1$ such that $|P_i'| = k$, $|P_i \cap P_i'| = k - g_{E_1}(k)$ and $E_1(P_i') = y'$. This implies we can find a set $P'_{\mathsf{flat}} \subset \mathscr{X}_1$ such that $|P'_{\mathsf{flat}}| = nk$, $|P_{\mathsf{flat}} \cap P'_{\mathsf{flat}}| = nk - f_{E_2}(n)f_{E_1}(k)$ and $\widetilde{E}(P'_{\mathsf{flat}}) = y$, i.e. we only need to change $f_{E_2}(n)f_{E_1}(k)$ points in P_{flat}, and $\widetilde{E}$ can become any value. So, we have

$$f_{\widetilde{E}}(nk) \leq f_{E_2}(n)f_{E_1}(k). \tag{3.25}$$

Applying Theorem 3.1 to E_1 and E_2, we obtain

$$g_{E_2}(n)g_{E_1}(k) \leq g_{\widetilde{E}}(nk). \tag{3.26}$$

Since $g_{\widetilde{E}}(nk) < f_{\widetilde{E}}(nk)$, from (3.25) and (3.26), we have

$$\frac{g_{E_2}(n)}{n}\frac{g_{E_1}(k)}{k} \leq \frac{g_{\widetilde{E}}(nk)}{nk} < \frac{f_{\widetilde{E}}(nk)}{nk} \leq \frac{f_{E_2}(n)}{n}\frac{f_{E_1}(k)}{k}. \tag{3.27}$$

Letting n, k go to infinity in (3.27), from (3.23) we obtain the breakdown point of $\widetilde{E}$ is

$$\tilde{\beta} = \lim_{n,k\to\infty} \frac{g_{\widetilde{E}}(nk)}{nk} = \lim_{n,k\to\infty} \frac{f_{\widetilde{E}}(nk)}{nk} = \beta_1\beta_2.$$

Since $E(P_{\mathsf{flat}}) = E_3(\widetilde{E}(\widetilde{P}^1_{\mathsf{flat}}), \widetilde{E}(\widetilde{P}^2_{\mathsf{flat}}), \cdots, \widetilde{E}(\widetilde{P}^m_{\mathsf{flat}}))$, we apply Theorem 3.2 to $\widetilde{E}$ and E_3, and then obtain (3.24). □

3.3 Applications

We next discuss several applications of our main theorems and observations. Applications 2 and 4 are direct applications of the easy to use theorems. Applications 1 and 3 take advantage of some of the nuances in definition, in particular the unexpected robustness of composing quantile estimators.

3.3.1 Application 1 : Balancing Percentiles

For n companies, for simplicity, assume each company has k employees. We are interested in the income of the regular employees of all companies, not the executives who may have

exorbitant pay. Let $p_{i,j}$ represents the income of the jth employee in the ith company. Set $P_{\mathsf{flat}} = \uplus_{i=1}^{n} P_i$ where the ith company has a set $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\} \subset \mathbb{R}$ and for notational convenience $p_{i,1} \leq p_{i,2} \leq \cdots \leq p_{i,k}$ for $i \in \{1, 2, \cdots, n\}$. Suppose the income data P_i of each company is preprocessed by a 45-percentile estimator E_1 (median of lowest 90% of incomes), with breakdown point $\beta_1 = 0.45$. In theory $E_1(P_i)$ can better reflect the income of regular employees in a company, since there may be about 10% of employees in the management of a company and their incomes are usually much higher than that of common employees. So, the preprocessed data is $X = \{E_1(P_1), E_1(P_2), \cdots, E_1(P_n)\}$.

If we define $E_2(X) = \text{median}(X)$ and $E(P_{\mathsf{flat}}) = E_2(X)$, then the breakdown point of E_2 is $\beta_2 = 0.5$, and the breakdown points of E is $\beta = \beta_1\beta_2 = 0.225$.

However, if we use another E_2, then E can be more robust. For example, for $X = \{x_1, x_2, \cdots, x_n\}$ where $x_1 \leq x_2 \leq \cdots \leq x_n$, we can define E_2 as the 55-percentile estimator (median of largest 90% of incomes). In order to make $E(P_{\mathsf{flat}}) = E_2(X) = E_2(E_1(P_1), E_1(P_2), \cdots, E_1(P_n))$ go to infinity, we need to either move 55% points of X to $-\infty$ or move 45% points of X to $+\infty$. In either case, we need to move about $0.45 \cdot 0.55nk$ points of P_{flat} to infinity. This means the breakdown point of E is $\beta = 0.45 \cdot 0.55 = 0.2475$ which is greater than 0.225.

This example implies if we know how the raw data is preprocessed by estimator E_1, we can choose a proper estimator E_2 to make the E_1-E_2 estimator more robust.

3.3.2 Application 2 : Regression of L_1 Medians

Suppose we want to use linear regression to robustly predict the weight of a person from his or her height, and we have multiple readings of each person's height and weight. The raw data is $P_{\mathsf{flat}} = \uplus_{i=1}^{n} P_i$ where for the ith person we have a set $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$ and $p_{i,j} = (x_{i,j}, y_{i,j})$ for $i \in \{1, 2, \cdots, n\}, j \in \{1, 2, \cdots, k\}$. Here, $x_{i,j}$ and $y_{i,j}$ are the height and weight respectively of the ith person in their jth measurement.

One "robust" way to process this data, is to first pre-process each P_i with its L_1-median [7]: $(\bar{x}_i, \bar{y}_i) \leftarrow E_1(P_i)$, where $E_1(P_i) = L_1\text{-median}(P_i)$ has breakdown point $\beta_1 = 0.5$. Then we could generate a linear model to predict weight $\hat{y}_i = ax + b$ from the Siegel Estimator [81]: $E_2(Z) = (a, b)$, with breakdown point $\beta_2 = 0.5$. From Theorem 3.2 we immediately know the breakdown point of $E(P_{\mathsf{flat}}) = E_2(E_1(P_1), E_1(P_2), \cdots, E_1(P_n))$ is $\beta = \beta_1\beta_2 = 0.5 \cdot 0.5 =$

0.25.

Alternatively, taking the Siegel estimator of P_{flat} (i.e., returning $E_2(P_{\mathsf{flat}})$) would have a much larger breakdown point of 0.5. So a seemingly harmless operation of normalizing the data with a robust estimator (with optimal 0.5 breakdown point) drastically decreases the robustness of the process.

3.3.3 Application 3 : Significance Thresholds

Suppose we are studying the distribution of the wingspread of fruit flies. There are $n = 500$ flies, and the variance of the true wingspread among these flies is on the order of 0.1 units. Our goal is to estimate the 0.05 significance level of this distribution of wingspread among normal flies.

To obtain a measured value of the wingspread of the ith fly, denoted F_i, we measure the wingspread of ith fly $k = 100$ times independently, and obtain the measurement set $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\}$. The measurement is carried out by a machine automatically and quickly, which implies the variance of each P_i is typically very small, perhaps only 0.0001 units, but there are outliers in P_i with small chance due to possible machine malfunction. This malfunction may be correlated to individual flies because of anatomical issues, or it may have autocorrelation (the machine jams for a series of consecutive measurements).

To perform hypothesis testing we desire the 0.05 significance level, so we are interested in the 95th percentile of the set $F = \{F_1, F_2, \cdots, F_n\}$. So a post processing estimator E_2 returns the 95th percentile of F and has a breakdown point of $\beta_2 = 0.05$ [54]. Now, we need to design an estimator E_1 to process the raw data $P_{\mathsf{flat}} = \uplus_{i=1}^{n} P_i$ to obtain $F = \{F_1, F_2, \cdots, F_n\}$. For example, we can define E_1 as $F_i = E_1(P_i) = \text{median}(P_i)$ and estimator E as $E(P_{\mathsf{flat}}) = E_2(E_1(P_1), E_1(P_2), \cdots, E_1(P_n))$.

Then, the breakdown point of E_1 is 0.5. Since the breakdown point of E_2 is 0.05, the breakdown point of the composite estimator E is $\beta = \beta_1\beta_2 = 0.5 \cdot 0.05 = 0.025$. This means if the measurement machine malfunctioned only 2.5% of the time, we could have an anomalous significant level, leading to false discovery. Can we make this process more robust by adjusting E_1?

Actually, *yes!*, we can use another pre-processing estimator to get a more robust E. Since the variance of each P_i is only 0.0001, we can let E_1 return the 5th percentile of a ranked

set of real numbers, then there is not much difference between $E_1(P_i)$ and the median of P_i. (Note: this introduces a small amount of bias that can likely be accounted for in other ways.) In order to make $E(P_{\mathsf{flat}}) = E_2(F)$ go to infinity we need to move 5% points of X to $-\infty$ (causing E_2 to give an anomalous value) or 95% points of X to $+\infty$ (causing many, 95%, of the E_1 values, to give anomalous values). In either case, we need to move about $5\% \cdot 95\%$ points of P_{flat} to infinity. So, the breakdown points of E is $\beta = 0.05 \cdot 0.95 = 0.0475$ which is greater than 0.025. That is, we can now sustain up to 4.75% of the measurement machine's reading to be anomalous, almost double than before, without leading to an anomalous significance threshold value.

This example implies if we know the post-processing estimator E_2, we can choose a proper method to preprocess the raw data to make the E_1-E_2 estimator more robust.

Remark 3.5. *A further study would be required to use such a composite estimator in practice due some bias it introduces. To replicate the normalization process on new experimental data (e.g., on a new species with hypothesized long wingspread), we would probably need to make one of the following adjustments to the standard process of measuring the wingspread of the new species and directly comparing it to the significance threshold. (a) Also consider the* 5*th percentile of the experimental measurements (with breakdown point* 0.05 *instead of* 0.5*). (b) Adjust the significance level by roughly* 0.0001 *units (the variance over P_i) making it conservative with respect to the* 5*th percentile versus the* 50*th percentile decision of each fly's measurements, so the* 50*th percentile could be used on the new experimental data. Or, (c) use a different percentile (say the* $(95+\varepsilon)$*th percentile instead of* 95*th) to balance the bias in using the* 5*th percentile of measurements. In the specific scenario we describe, we believe option (b) may be a very acceptable option with little lack in precision (due to difference in variance* 0.1 *and* 0.0001*) but with large gain in robustness.*

3.3.4 Application 4 : 3-Level Composition

Suppose we want to use a single value to represent the temperature of the US in a certain day. There are $m = 50$ states in the country. Suppose each state has $n = 100$ meteorological stations, and the station i in state j measures the local temperature $k = 24$ times to get the data $P_{i,j} = \{t_{i,j,1}, t_{i,j,2}, \cdots, t_{i,j,k}\}$. We define $P^j_{\mathsf{flat}} = \uplus_{i=1}^{n} P_{i,j}$, $P_{\mathsf{flat}} = \uplus_{j=1}^{m} P^j_{\mathsf{flat}}$ and

$$E_1(P_{i,j}) = \text{median}(P_{i,j}), \quad E_2(P^j_{\mathsf{flat}}) = \text{median}\left(E_1(P_{1,j}), E_1(P_{1,j}), \cdots, E_1(P_{n,j})\right)$$

$$E(P_{\mathsf{flat}}) = E_3(E_2(P^1_{\mathsf{flat}}), E_2(P^2_{\mathsf{flat}}), \cdots, E_2(P^m_{\mathsf{flat}})) = \text{median}(E_2(P^1_{\mathsf{flat}}), E_2(P^2_{\mathsf{flat}}), \cdots, E_2(P^m_{\mathsf{flat}})).$$

So, the break down points of E_1, E_2 and E_3 are $\beta_1 = \beta_2 = \beta_3 = 0.5$. From Theorem 3.3, we know the break down point of E is $\beta = \beta_1\beta_2\beta_3 = 0.125$. Therefore, we know the estimator E is not very robust, and it may be not a good choice to use $E(P_{\mathsf{flat}})$ to represent the temperature of the US in a certain day.

This example illustrates how the more times the raw data is aggregated, the more unreliable the final result can become.

3.4 Simulations

We next describe a few more scenarios where our new theory on estimator composition is relevant. For these we simulate a couple of data sets to demonstrate how one might construct interesting algorithms from these ideas.

3.4.1 Simulation 1 : Estimator Manipulation

In this simulation we actually construct a method to relocate an estimator by modifying the smallest number of points possible. We specifically target the L_1-median of L_1-medians since its somewhat non-trivial to solve for the new location of data points.

In particular, given a target point $p_0 \in \mathbb{R}^2$ and a set of nk points $P_{\mathsf{flat}} = \uplus_{i=1}^n P_i$, where $P_i = \{p_{i,1}, p_{i,2}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$, we use simulation to show that we only need to change $\tilde{n}\tilde{k}$ points of P_{flat}, then we can get a new set $\widetilde{P}_{\mathsf{flat}} = \uplus_{i=1}^n \widetilde{P}_i$ such that $\text{median}(\text{median}(\widetilde{P}_1), \text{median}(\widetilde{P}_2), \cdots, \text{median}(\widetilde{P}_n)) = p_0$. Here, the "median" means L_1-median, and

$$\tilde{n} = \begin{cases} \frac{1}{2}n & \text{if } n \text{ is even} \\ \frac{1}{2}(n+1) & \text{if } n \text{ is odd} \end{cases}, \quad \tilde{k} = \begin{cases} \frac{1}{2}k & \text{if } k \text{ is even} \\ \frac{1}{2}(k+1) & \text{if } k \text{ is odd} \end{cases}.$$

To do this, we first show that, given k points $S = \{(x_i, y_i) \mid 1 \le i \le k\}$ in $\mathbb{R}^2$, and a target point (x_0, y_0), we can change $\tilde{k}$ points of S to make (x_0, y_0) as the L_1-median of the new set. As n and k grow, then $\tilde{n}\tilde{k}/(nk) = 0.25$ is the asymptotic breakdown point of this estimator, as a consequence of Theorem 3.2, and thus we may need to move this many points to get the result.

If (x_0, y_0) is the L_1-median of the set $\{(x_i, y_i) \mid 1 \le i \le k\}$, then we have [89]:

$$\sum_{i=1}^k \frac{x_i - x_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} = 0, \ \sum_{i=1}^k \frac{y_i - y_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} = 0. \tag{3.28}$$

We define $\vec{x} = (x_1, x_2, \cdots, x_{\tilde{k}})$, $\vec{y} = (y_1, y_2, \cdots, y_{\tilde{k}})$ and

$$h(\vec{x}, \vec{y}) = \left(\sum_{i=1}^{k} \frac{x_i - x_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right)^2 + \left(\sum_{i=1}^{k} \frac{y_i - y_0}{\sqrt{(x_i - x_0)^2 + (y_i - y_0)^2}} \right)^2.$$

Since (3.28) is the sufficient and necessary condition for L_1-median, if we can find $\vec{x}$ and $\vec{y}$ such that $h(\vec{x}, \vec{y}) = 0$, then (x_0, y_0) is the L_1-median of the new set.

Since

$$\begin{aligned}
\partial_{x_i} h(\vec{x}, \vec{y}) =& 2\Big(\sum_{j=1}^{k} \frac{x_j - x_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}} \Big) \frac{(y_i - y_0)^2}{\big((x_i - x_0)^2 + (y_i - y_0)^2\big)^{\frac{3}{2}}} \\
& - 2\Big(\sum_{j=1}^{k} \frac{y_j - y_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}} \Big) \frac{(x_i - x_0)(y_i - y_0)}{\big((x_i - x_0)^2 + (y_i - y_0)^2\big)^{\frac{3}{2}}}, \\
\partial_{y_i} h(\vec{x}, \vec{y}) =& -2\Big(\sum_{j=1}^{k} \frac{x_j - x_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}} \Big) \frac{(x_i - x_0)(y_i - y_0)}{\big((x_i - x_0)^2 + (y_i - y_0)^2\big)^{\frac{3}{2}}} \\
& + 2\Big(\sum_{j=1}^{k} \frac{y_j - y_0}{\sqrt{(x_j - x_0)^2 + (y_j - y_0)^2}} \Big) \frac{(x_i - x_0)^2}{\big((x_i - x_0)^2 + (y_i - y_0)^2\big)^{\frac{3}{2}}},
\end{aligned}$$

we can use gradient descent to compute $\vec{x}, \vec{y}$ to minimize h. For the input $S = \{(x_i, y_i) | 1 \leq i \leq k\}$, we choose the initial value $\vec{x}_0 = \{x_1, x_2, \cdots, x_{\tilde{k}}\}$, $\vec{y}_0 = \{y_1, y_2, \cdots, y_{\tilde{k}}\}$, and then update $\vec{x}$ and $\vec{y}$ along the negative gradient direction of h, until the Euclidean norm of gradient is less than 0.00001.

The algorithm framework is then as follows, using the above gradient descent formulation at each step. We first compute the L_1-median m_i for each P_i, and then change $\tilde{n}$ points in $\{m_1, m_2, \cdots, m_n\}$ to obtain

$$\{m'_1, m'_2, \cdots, m'_{\tilde{n}}, m_{\tilde{n}+1}, \cdots, m_n\}$$

such that $\text{median}(m'_1, m'_2, \cdots, m'_{\tilde{n}}, m_{\tilde{n}+1}, \cdots, m_n) = p_0$. For each m'_i, we change $\tilde{k}$ points in P_i to obtain

$$\widetilde{P}_i = \{p'_{i,1}, p'_{i,2}, \cdots, p'_{i,\tilde{k}}, p_{i,\tilde{k}+1}, \cdots, p_{i,k}\}$$

such that $\text{median}(\widetilde{P}_i) = m'_i$. Thus, we have

$$\text{median}\big(\text{median}(\widetilde{P}_1), \cdots, \text{median}(\widetilde{P}_{\tilde{n}}), \text{median}(P_{\tilde{n}+1}), \cdots, \text{median}(P_n)\big) = p_0. \tag{3.29}$$

To show a simulation of this process, we use a uniform distribution to randomly generate nk points in the region $[-10, 10] \times [-10, 10]$, and generate a target point $p_0 = (x_0, y_0)$ in the

Table 3.1: The running result of Simulation 1.

n	k	$\tilde{n}$	$\tilde{k}$	(x_0, y_0)	(x_0', y_0')
5	8	3	4	(0.9961, 1.0126)	(0.9961, 1.0126)
5	8	3	4	(10.7631, 11.0663)	(10.7025 11.0623)
10	5	5	3	(-13.8252, -4.7462)	(-13.8330, -4.7482)
50	20	25	10	(-14.7196, -13.6728)	(-14.7263, -13.6784)
100	50	50	25	(-14.0778, 18.3665)	(-14.0773, 18.3658)
500	100	250	50	(-15.8408, -6.4259)	(-15.8385, -6.4250)
1000	200	500	100	(18.6351, -12.1014)	(18.7886, -12.2011)

region $[-20, 20] \times [-20, 20]$, and then use our algorithm to change $\tilde{n}\tilde{k}$ points in the given set, to make the new set satisfy (3.29). Table 3.1 shows the result of running this experiment for different n and k, where (x_0', y_0') is the median of medians for the new set obtained by our algorithm. It lists the various values n and k, the corresponding values $\tilde{n}$ and $\tilde{k}$ of points modified, and the target point and result of our algorithm. If we reduce the terminating condition, which means increasing the number of iteration, we can obtain a more accurate result, but only requiring the Euclidean norm of gradient to be less than 0.00001, we get very accurate results, within about 0.01 in each coordinate.

We illustrate the results of this process graphically for a couple of examples in Table 3.1; for the cases $n = 5$, $k = 8$, $(x_0, y_0) = (0.9961, 1.0126)$ and $n = 5$, $k = 8$, $(x_0, y_0) = (10.7631, 11.0663)$ These are shown in Figure 3.1 and Figure 3.2, respectively. In these two figures, the green star is the target point. Since $n = 5$, we use five different markers (circle, square, upward-pointing triangle, downward-pointing triangle, and diamond) to represent five kinds of points. The given data P_{flat} are shown by black points and unfilled points. Our algorithm changes those unfilled points to the blue ones, and the green points are the medians of the new subsets. The red star is the median of medians for P_{flat}, and other red points are the median of old subsets. So, we only changed 12 points out of 40, and the median of medians for the new data set is very close to the target point.

3.4.2 Simulation 2 : Router Monitoring

Suppose there are $n = 100$ routers in a network, and each router monitors a stream of length $k = 1000$. A router can use streaming algorithm to monitor a single percentile, for instance the frugal algorithm here [70] only needs a few bites per percentile maintained – it

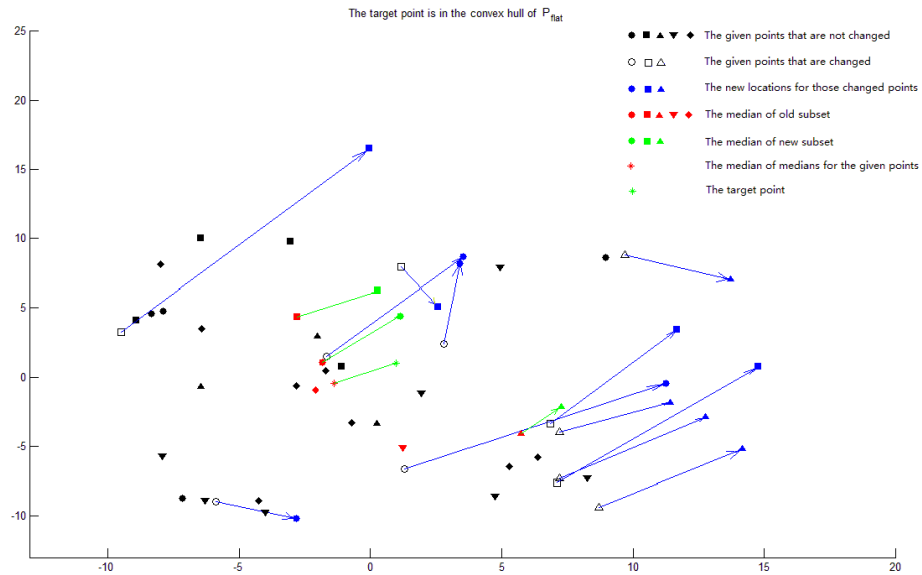

Figure 3.1: The running result for the case $n = 5$, $k = 8$, $(x_0, y_0) = (0.9961, 1.0126)$ in Table 3.1.

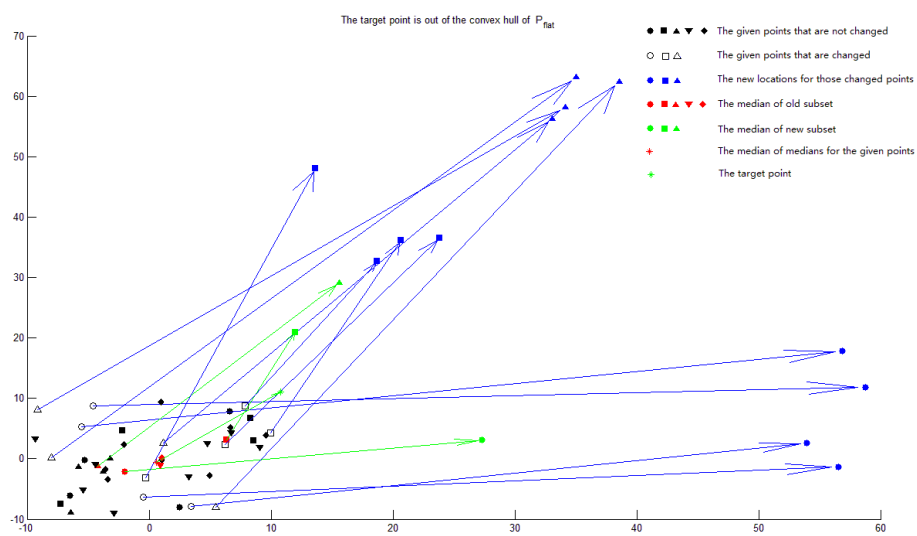

Figure 3.2: The running result for the case $n = 5$, $k = 8$, $(x_0, y_0) = (10.7631, 11.0663)$ in Table 3.1.

does not need to monitor all. We will consider monitoring the approximate median (50% percentile), 10% percentile, and 90% percentile of the stream, and sending these to a single command center. The command center will analyze these data to determine whether an attack occurs. In practice, command centers monitor much larger streams (values of k) and many more routers (values of n).

Table 3.2: The output for different combinations of estimators and outliers.

Proportion of outliers	location of outliers	n_1	k_1	E_1: 10% E_2: 10%	E_1: 90% E_2: 90%	E_1: 10% E_2: 90%	E_1: 90% E_2: 10%	E_1: median E_2: median
0%		0	0	-1.3327	1.3549	-1.2169	1.2254	-0.0085
1.21%	[100,110]	11	110	-1.3539	**100.5666**	-1.2033	1.2093	0.0091
1.21%	[-110,-100]	11	110	**-100.6573**	1.3291	-1.2065	1.2175	0.0021
10.01%	[100,110]	11	910	-1.3364	**108.6957**	**100.0553**	1.2118	0.0082
10.01%	[-110,-100]	11	910	**-108.7768**	1.3388	-1.2081	**-100.0721**	-0.0119
26.01%	[100,110]	51	510	-1.3388	**108.1641**	-0.7794	1.2347	**100.1062**
26.01%	[-110,-100]	51	510	**-108.2083**	1.3163	-1.2313	0.7697	**-100.1018**
46.41%	[100,110]	51	910	-1.3350	**108.9832**	**100.1411**	1.2280	**104.2258**
46.41%	[-110,-100]	51	910	**-109.0043**	1.3350	-1.2423	**-100.1340**	**-104.0705**

We use standard normal distribution to generate an array S_i with 1000 entries to simulate the ith stream, and assume the routers use the estimator E_1 to process streams, i.e. E_1 returns the approximate 10% percentile, or 90% percentile, or the median of a stream. The command center uses the estimator E_2 to process the gathered data $S = (E_1(S_1), E_1(S_2), \cdots, E_1(S_n))$, and E_2 can return the 10% percentile, or 90% percentile, or the median of S. In our simulation, we compute each of these quantities exactly. We use outliers in interval $[100, 110]$ or $[-110, -100]$ to simulate attacks.

These values may represent some statistic deemed worth monitoring, say the packet length or header size after it has been appropriately normalized.

We choose n_1 streams, and put k_1 outliers from the same interval (all positive, or all negative) to each chosen stream. Table 3.2 shows the final output from command center for different combinations of estimators and outliers. The first column in Table 3.2 shows the proportion of outliers, which is equal to $\frac{n_1 k_1}{nk}$. For example, in the third row of the table, we choose 11 streams randomly and put 110 outliers drawn from [100,110] into each chosen stream, so the proportion of outliers is $(11 \times 110)/(100 \times 1000) = 1.21\%$. When a value being monitored as a composite of various percentiles becomes very large (above 100, so not from the normal distribution) we mark it **bold**.

It is shown in Table 3.2 that for the case $E_1 : 10\%$, $E_2 : 10\%$ and $E_1 : 90\%$, $E_2 : 90\%$, we can use 1.21% of outliers to change the output of E_1-E_2 estimator, since in this situation the breakdown point of E_1-E_2 estimator is 0.01. For the case $E_1 : 10\%, E_2 : 90\%$ and $E_1 : 10\%, E_2 : 90\%$, we can use 10.01% of outliers to change the output of E_1-E_2 estimator, since in this situation the breakdown point of E_1-E_2 estimator is 0.09. When E_1 and E_2 both

return the median of a data set, we can use 26.01% of outliers to change the output of E_1-E_2 estimator, since in this situation the breakdown point of E_1-E_2 estimator is 0.25.

This experiment illustrates how using various composite estimators with different percentiles can highlight various levels of potential distributed denial of service attacks. For instance, if only the $E_1 : 10\%, E_2 : 10\%$ estimator is flagged, then we see a few routers have a few anomalous packets, and even though it is distributed to only about 10% of routers and 10% of data, we can observe it; but for the most part would be at most a warning. If $E_1 : 10\%, E_2 : 90\%$ estimator or $E_1 : 50\%, E_2 : 50\%$ estimator is flagged, it means at least 9% or 25% of the packets across all routers much be anomalous, and we may see a real DDS or an early sign of one. These are all conservative estimates. On the other hand, if at least 10% of the packets are modified on 10% of routers (not too much, perhaps as little as 1%), then the $E_1 : 10\%, E_2 : 10\%$ estimator will definitely observe it. And if at least 10% of the packets are modified on 50% of the routers (over 5% of all packets), then an $E_1 : 10\%, E_2 : 50\%$ estimator will definitely observe it. Further work is required to discover the best combination of percentiles to monitor, but using our observations about composite estimators suggests this approach which can monitor against various distributions of DDS attacks without only a few simple estimators, requiring a few bites each, at each router.

3.5 Discussion

In this chapter, we define the breakdown point of the composition of two or more estimators. These definitions are technical but necessary to understand the robustness of composite estimators; and they do not stray too far from prior formal definitions [32, 52]. Generally, the composition of two or more estimators is less robust than each individual estimator. We highlight a few applications and believe many more exist. These results already provide important insights for complex data analysis pipelines common to large-scale automated data analysis. Moreover, these approaches provides worst case guarantees that are concrete about when outliers can or cannot create a problem, as opposed to some regularization-based approaches that just tend to work on most data.

Next we will highlight a few more insights from this work, or discuss challenges for follow-on work.

On the dangers of composition. The common case of composing two estimators, each

with breakdown point of 0.5 yields a composite estimator of 0.25. This means if the result is anomalous, at least 25% of the data must change, down from 50%. In other cases, the resulting composite estimator might yield an even smaller breakdown point of say 0.05. This seems like very bad news! But for large data sets, adversarially changing 5% of data is still a lot. For instance with 1 million data points, then 5% is $50,000$, which would still be an ominously difficult task to modify. So even a 0.05 or 0.01 breakdown point on large data is a useful barrier to manipulation (of the sort in our Simulation 1 below). On the other hand, repeated composition can quickly (exponentially) decrease the breakdown point until it is dangerously low; hence we believe this new theory will play an import role in understanding the robustness and security of long data analysis pipelines.

Robustness and unbiasedness. In this chapter, we focus exclusively on the robustness of estimators, but it is also important to aim for low-MSE or unbiasedness estimators. An interesting future direction is to design estimators that are both robust (including have large onto-breakdown points) as well as other properties. We lead this direction with a few points:

- Composing two unbiased estimators will typically be unbiased (some care may be needed in weighting).
- Robustness is a worst-case analysis (protecting against adversarial data) and its claims are often orthogonal to those about low-MSE.
- Our analysis bounds the robustness of composition of *any* two (or more) estimators. So if other work independently shows low-MSE or low-bias properties, then we can immediately combine these works to show both.

Removing all subsets size k constraint. The restriction $|P_i| = k$ (all subsets at the first level are the same size) is mainly for expositional convenience. Otherwise, there are some technical issues with reweighing points in P_{flat} and defining the limits. In fact, suppose $|P_i| = k_i$ for $i = 1, 2, \cdots, n$, $P_{\mathsf{flat}} = \uplus_{i=1}^{n} P_i$, $g_{E_1}(k_1) \leq g_{E_1}(k_2) \leq \cdots \leq g_{E_1}(k_n)$, and

$$E(P_{\mathsf{flat}}) = E_2\left(E_1(P_1), E_1(P_2), \cdots, E_1(P_n)\right).$$

Then using the method in the proof of Theorem 3.1, we can obtain a result similar :

$$\sum_{i=1}^{g_{E_2}(n)} g_{E_1}(k_i) \leq g_E(\sum_{i=1}^{n} k_i) \tag{3.30}$$

which is a generalization of (3.10).

Finite sampling breakdown point for composite estimators. Theorem 3.2 provides an asymptotic breakdown point for composite estimators. But for smaller data sets, a finite sample version is also useful and important. Equation (3.10) already gives a lower bound of the finite sample breakdown point of composite estimators. To get an upper bound on the finite sample vesion, we can modify Theorem 3.2, by adding a condition $f_{E_1}(k) = g_{E_1}(k) + C$ where C is a positive constant. Then there is also an annoying off-by-one error on g_{E_2} (see eq (3.20)), so the result would be something like

$$g_{E_1}(k)g_{E_2}(n) \leq g_E(nk) \leq (g_{E_1}(k) + C)(g_{E_2}(n) + 1),$$

and it is not completely tight. We leave providing a tight bound (up to these constants) as an open question.

CHAPTER 4

SIMPLE DISTANCES FOR TRAJECTORIES VIA LANDMARKS

4.1 Introduction

The *choice* of a distance is often the most important modeling decision in any data analysis task. This choice is what determines which objects are close and which are far. However, this task is often taken lightly or made just based on what provides the simplest or easiest to compute option.

In this chapter, we explore what we believe to be a new and natural family of distances between objects, focusing on two cases when the objects are hyperplanes (e.g., regressors or separators), or when they are trajectories. Our proposed distance $\mathtt{d}_Q$ uses a set Q of landmark points, which could be the dataset that regressors or separators are trained on, or in the case of trajectories these may be points of interest for which a trajectory passing nearby has specific meaning. However, in a general case, Q can be chosen as arbitrary or random points placed to cover a domain of focus. Then the new distances, instead of being directly between the objects themselves, are based on how they interact with the set of landmarks. In the simplest variant, for n landmarks Q, for any object J we create an n-dimensional vector $v_J = (v_1, v_2, \ldots, v_n)$ of the distance from $q_i \in Q$ to J, and the distance between two objects J_1 and J_2 is the Euclidean distance between the vectors $\|v_{J_1} - v_{J_2}\|$. In other words, we *vectorize* the distance between complex objects.

In this chapter, we explore several variants of this formulation, derive convenient mathematical properties, and demonstrate its efficacy in several data analysis scenarios.

Key properties of a distance. A definition of a distance $\mathtt{d}$ is the key building block in most data analysis tasks. For instance, it is at the heart of any assignment-based clustering (e.g., k-means) or for nearest-neighbor searching and analysis. We can also define a radial-

basis kernel $K(p,q) = \exp(-\mathtt{d}(p,q)^2)$ (or similarly), which is required for kernel SVM classification, kernel regression, and kernel density estimation. A change in the distance, directly affects the meaning and modeling inherent in each of these tasks. So the first consideration in choosing a distance should always be, does it capture the properties between the objects that matter?

As we will observe, by having a distance depend on a set of landmarks Q, then we can tune it to focus on certain regions. In the case of regressors or separators (e.g., infinite lines, hyperplanes) this makes sure the distance is determined by how these infinite objects interact with the support of the data. In the case of trajectories, the distance can be adjusted to focus on one or more locations of interest (e.g., a sporting event or school) or regions of interest (e.g., how someone passes through an airport, but not how they get there), as opposed to its full geometry.

A generic desired property of a distance is that it should be a metric: for instance this is essential in the analysis for the Gonzalez algorithm [49] for k-center clustering, and many other contexts such as nearest-neighbor searching.

Another generic goal is analyzing the distance's metric balls. That is, given a set of objects $\mathcal{J}$ and a distance $\mathtt{d} : \mathcal{J} \times \mathcal{J} \to \mathbb{R}$, let $B(J,r) = \{J' \in \mathcal{J} \mid \mathtt{d}(J,J') \leq r\}$ be a metric ball around $J \in \mathcal{J}$ of radius r. Then we can define a range space $(\mathcal{J}, \mathcal{R})$ where $\mathcal{R} = \{B(J,r) \mid J \in \mathcal{J}, r \geq 0\}$, and consider its VC-dimension [85]. When the VC-dimension ν is small, it implies that the metric balls cannot interact with each other in a too complex way, indicating the distance is roughly as well-behaved as a ν-dimensional Euclidean ball. More directly, this implies, decision boundaries to classify objects can be learned with only ε-fraction generalization error using $O(\nu/\varepsilon \cdot \log(1/\varepsilon))$ samples if the data is separable, or $O(\nu/\varepsilon^2)$ samples if the data is not separable [65]. Similar bounds can be shown for other tasks such as preserving kernel density estimates derived from such distances [61]. In other words, this ensures that many tasks are stable with respect to the underlying family of objects $\mathcal{J}$.

Main results. We define a new data dependent distance $\mathtt{d}_Q$ for trajectories and for linear models (e.g., regressors, separators) built from a landmark data set Q. For the simpler cases of linear models (in Section 4.2), we show it is a metric as long as Q is full rank. We also show that its metric balls have VC-dimension bounded only by the ambient dimension

and not on the size of Q. We find this surprising because the distance corresponds to an embedding in $|Q|$-dimensional Euclidean space where an immediate bound for the VC-dimension is $|Q| + 1$; and indeed this will be the best bound we have for most of the trajectory variants. We show how to directly extend all of these definitions of lines to trajectories, with a somewhat unintuitive and restrictive distance measure $\mathtt{d}_Q^{\leftrightarrow}$.

For the pressing scenario of trajectories, in Section 4.3, we introduce two more intuitive variants $\mathtt{d}_Q$ and $\mathtt{d}_Q^{\pi}$. We describe simple conditions for Q under which they are metrics. We can immediately see that both distances are pseudometrics (they satisfy triangle inequality, and are symmetric, but might have distinct objects with distance 0). We show they satisfy the final 0-property of a metric as long as the waypoints are distinct and Q is sufficiently dense. For all new variants we demonstrate that they are at the least as effective for classification tasks (via KNN classifiers) as compared to the best of 9 other common metrics, and *in some cases significantly outperforms all of these measures*. Moreover, the previous competing variants are typically significantly more complicated or computationally intensive, and may require parameter tuning.

In contrast to most of these trajectory distance alternatives, all of our proposed distances are very simple to compute and work with. They map curves (or hyperplanes) to a $|Q|$-dimensional parameter space where Euclidean distance (or similar) is used. In $\mathtt{d}_Q$ for curves, each coordinate v_i is the distance to the closest point on the curve from $q_i \in Q$. In $\mathtt{d}_Q^{\pi}$ each "coordinate" is actually the d coordinates of the closest point on the curve (not just the distance). In $\mathtt{d}_Q^{\leftrightarrow}$ each "coordinate" v_i is actually k values, to the distance to the closest point on the k lines extending the k lines segments of the curve. These mappings are effective with only 10 or 20 landmark points Q. And because they have a familiar Euclidean structure, we can immediately invoke favorite algorithms in this space, from Lloyds for k-means clustering, linear and kernel SVM, and highly-engineered approximate nearest neighbor libraries. In comparison to recent trajectory similarity search systems [80, 92], we show using $\mathtt{d}_Q$ is much simpler and several orders of magnitude faster.

In summary, this chapter introduces a family of metrics for regressors, separators, and piecewise-linear curves which are incredibly simple to use, provide a sketch vector in Euclidean space, have many other desirable mathematical properties, and perform as well as and often significantly better than any existing measure.

4.2 Distance Between Lines and Hyperplanes

As a warm up to the general case, we define a new landmark-based distance $\mathtt{d}_Q$ between two lines, and give the condition under which it is a metric. Then we generalize to hyperplanes, and provide the general metric proof, the VC-dimension of metric ball proof, and some algorithmic implications. We conclude with a direct extension to trajectories.

4.2.1 Warm Up: Distance Between Lines

We begin by reviewing alternatives, starting with the default *dual Euclidean distance*. Consider the least square regression problem in $\mathbb{R}^2$: given $Q = \{(x_1, y_1), \cdots, (x_n, y_n)\} \subset \mathbb{R}^2$, return a line $\ell : y = ax + b$ such that $(a, b) = \arg\min_{(a,b)\in\mathbb{R}^2} \sum_{i=1}^{n} (ax_i + b - y_i)^2$. If $\ell_1 : y = a_1 x + b_1$ is an alternate fit to this data, then to measure the difference in these variants, we can define a distance between ℓ and ℓ_1. A simple and commonly used distance (which we called the *dual-Euclidean distance*) is

$$\mathtt{d}_{\mathsf{dE}}(\ell, \ell_1) := \sqrt{(a - a_1)^2 + (b - b_1)^2}.$$

This can be viewed as dualizing the lines into a space defined by their parameters (slope a and intercept b), and then taking the Euclidean distance between these parametric points. However, as shown in Figure 4.1(Left), if both ℓ_1 and ℓ_2 have the same slope $a_1 = a_2$, and are offset the same amount from ℓ ($|b - b_1| = |b - b_2|$), then $\mathtt{d}_{\mathsf{dE}}(\ell, \ell_1) = \mathtt{d}_{\mathsf{dE}}(\ell, \ell_2)$, although intuitively ℓ_1 does a much more similar job to ℓ with respect to Q than does ℓ_2.

More generically, a geometric object is usually described by an (often compact) set in $\mathbb{R}^d$. There are many ways to define and compute distances between such objects [8, 9, 47, 48]. These can be based on the minimum [47, 48] or maximum (e.g., Hausdorff) [8, 9] distance between objects. We review more later in the context of trajectories in Section 4.4.1. For lines or hyperplanes which extend infinitely and may intersect at single points, such measures are not meaningful.

Our formulation. Suppose $Q = \{q_1, q_2, \cdots, q_n\} \subset \mathbb{R}^2$ where q_i has coordinates (x_i, y_i) for $1 \leq i \leq n$, and ℓ is a line in $\mathbb{R}^2$, then ℓ can be uniquely expressed as

$$\ell = \{(x, y) \in \mathbb{R}^2 \mid u_1 x + u_2 y + u_3 = 0\},$$

where $(u_1, u_2, u_3) \in \mathbb{U}^3$. Here $\mathbb{U}^3 = \{u = (u_1, u_2, u_3) \in \mathbb{R}^3 \mid u_1^2 + u_2^2 = 1$ and the first nonzero entry of u is positive$\}$, is a canonical way to normalize u where (u_1, u_2) is unit

normal vector and u_3 is an offset parameter. Let $v_{Q_i}(\ell) = u_1 x_i + u_2 y_i + u_3$; it is the signed distance from $q_i = (x_i, y_i)$ to the closest point on ℓ. Then $v_Q(\ell) = (v_{Q_1}(\ell), v_{Q_2}(\ell), \ldots, v_{Q_n}(\ell))$ is the n-dimensional vector of these distances. For two lines ℓ_1, ℓ_2 in $\mathbb{R}^2$, we can now define

$$\mathtt{d}_Q(\ell_1, \ell_2) = \left\| \frac{1}{\sqrt{n}} (v_Q(\ell_1) - v_Q(\ell_2)) \right\| = \Big(\sum_{i=1}^{n} \frac{1}{n} (v_{Q_i}(\ell_1) - v_{Q_i}(\ell_2))^2 \Big)^{\frac{1}{2}}.$$

As shown in Figure 4.1(Right), $|v_{Q_i}(\ell)|$ is the distance from q_i to ℓ. With the help of Q, we convert each line ℓ in $\mathbb{R}^2$ to point $\frac{1}{\sqrt{n}} v_Q(\ell)$ in $\mathbb{R}^n$, and use the Euclidean distance between two points to define the distance between the original two lines. Via this Euclidean embedding, it directly follows that $\mathtt{d}_Q$ is symmetric and follows the triangle inequality. The following theorem shows, under reasonable assumptions of Q, no two different lines can be mapped to the same point in $\mathbb{R}^n$, so $\mathtt{d}_Q$ is a metric.

Theorem 4.1. *Suppose in $Q = \{(x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)\} \subset \mathbb{R}^2$ there are three non-collinear points, and $\mathcal{L} = \{\ell \mid \ell$ is a line in $\mathbb{R}^2\}$, then $\mathtt{d}_Q$ is a metric in $\mathcal{L}$.*

Proof. The function $\mathtt{d}_Q(\cdot, \cdot)$ is symmetric and by mapping to $\mathbb{R}^n$ satisfies the triangle inequality, and $\ell_1 = \ell_2$ implies $\mathtt{d}_Q(\ell_1, \ell_2) = 0$; we now show if $\mathtt{d}_Q(\ell_1, \ell_2) = 0$, then $\ell_1 = \ell_2$.

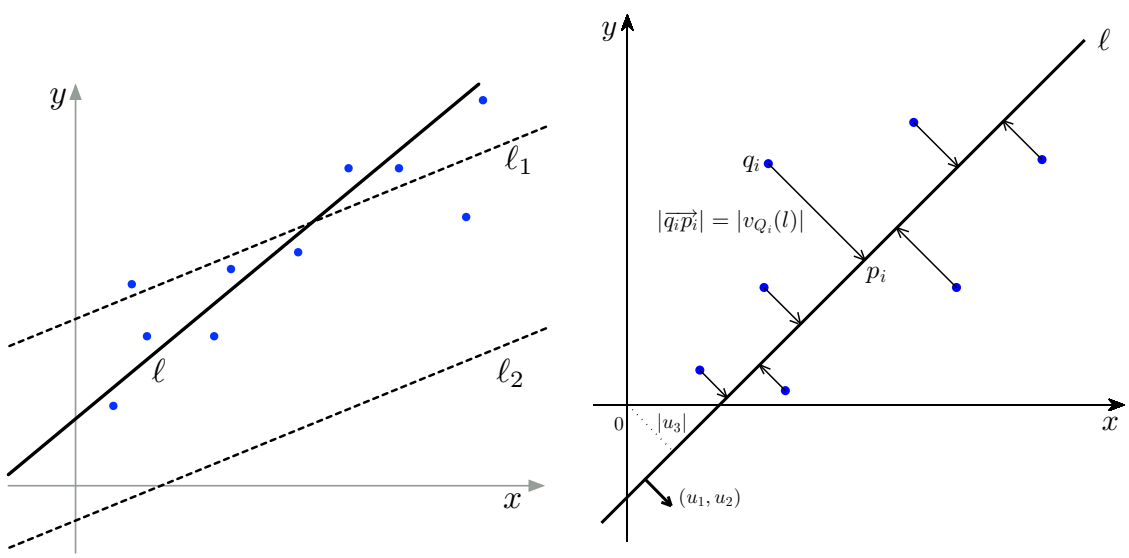

Figure 4.1: Left: $\mathtt{d}_{\mathtt{dE}}(\ell, \ell_1) = \mathtt{d}_{\mathtt{dE}}(\ell, \ell_2)$, but which of ℓ_1 and ℓ_2 is more similar to ℓ with respect to Q? Right: Each p_i is the projection of q_i on ℓ.

Without loss of generality, we assume (x_1, y_1), (x_2, y_2), $(x_3, y_3) \in Q$ are not on the same line, which implies

$$\begin{vmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{vmatrix} \neq 0. \tag{4.1}$$

Suppose ℓ_1 and ℓ_2 are expressed in the form:

$$\ell_1 = \{(x, y) \in \mathbb{R} \mid u_1^{(1)} x + u_2^{(1)} y + u_3^{(1)} = 0\},$$
$$\ell_2 = \{(x, y) \in \mathbb{R} \mid u_1^{(2)} x + u_2^{(2)} y + u_3^{(2)} = 0\},$$

where $(u_1^{(1)}, u_2^{(1)}, u_3^{(1)}), (u_1^{(2)}, u_2^{(2)}, u_3^{(2)}) \in \mathbb{U}^3$ represent lines ℓ_1 and ℓ_2, respectively. If $\mathtt{d}_Q(\ell_1, \ell_2) = 0$, then we have

$$x_i(u_1^{(1)} - u_1^{(2)}) + y_i(u_2^{(1)} - u_2^{(2)}) + (u_3^{(1)} - u_3^{(2)}) = 0$$

for $i = 1, 2, 3$. We can write this as the system

$$\begin{bmatrix} x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \\ x_3 & y_3 & 1 \end{bmatrix} \begin{bmatrix} u_1^{(1)} - u_1^{(2)} \\ u_2^{(1)} - u_2^{(2)} \\ u_3^{(1)} - u_3^{(2)} \end{bmatrix} = 0.$$

Using (4.1), we know it has the unique solution $[u_1^{(1)} - u_1^{(2)}, u_2^{(1)} - u_2^{(2)}, u_3^{(1)} - u_3^{(2)}]^T = [0, 0, 0]^T$. So, we have $u_1^{(1)} = u_1^{(2)}$, $u_2^{(1)} = u_2^{(2)}$ and $u_3^{(1)} =_3^{(2)}$, and thus $\ell_1 = \ell_2$. □

Remark 4.1. *In the above formulation, the absolute value $|v_{Q_i}(\ell)|$ is the distance from (x_i, y_i) to the line ℓ, i.e. $|v_{Q_i}(\ell)| = \min_{(x,y)\in\ell}((x - x_i)^2 + (y - y_i)^2)^{\frac{1}{2}}$. Moreover, if ℓ is parallel to ℓ', then $|v_{Q_i}(\ell) - v_{Q_i}(\ell')| = \min_{(x,y)\in\ell,(x',y')\in\ell'}((x - x')^2 + (y - y')^2)^{\frac{1}{2}}$ for any $i \in [n]$, which means $\mathtt{d}_Q$ is a generalization of the natural offset distance between two parallel lines.*

4.2.2 Distance Between Hyperplanes

Now let $\mathcal{H} = \{h \mid h \text{ is a hyperplane in } \mathbb{R}^d\}$ represent the space of all hyperplanes. Suppose $Q = \{q_1, q_2, \cdots, q_n\} \subset \mathbb{R}^d$, where q_i has the coordinate $(x_{i,1}, x_{i,2}. \cdots, x_{i,d})$. Any hyperplane $h \in \mathcal{H}$ can be uniquely expressed in the form

$$h = \{x = (x_1, \cdots, x_d) \in \mathbb{R}^d \mid \textstyle\sum_{j=1}^d u_j x_j + u_{d+1} = 0\},$$

where $(u_1, \cdots, u_{d+1})$ is a vector in $\mathbb{U}^{d+1} := \{u = (u_1, \cdots, u_{d+1}) \in \mathbb{R}^{d+1} \mid \sum_{j=1}^d u_j^2 = 1$ and the first nonzero entry of u is positive$\}$, i.e. $(u_1, \cdots, u_d)$ is the unit normal vector of

h, and u_{d+1} is the offset. We introduce the notation $v_Q(h) = (v_{Q_1}(h), \cdots, v_{Q_n}(h))$ where $v_{Q_i}(h)$ is again the signed distance from q_i to the closest point on h. We can specify $v_{Q_i}(h) = \sum_{j=1}^{d} u_j x_{i,j} + u_{d+1}$, which is a dot-product with the unit normal of h, plus offset u_{d+1}. Now for two hyperplanes h_1, h_2 in $\mathbb{R}^d$ define

$$\mathtt{d}_Q(h_1, h_2) := \Big\| \frac{1}{\sqrt{n}}(v_Q(h_1) - v_Q(h_2)) \Big\| = \Big(\sum_{i=1}^{n} \frac{1}{n}(v_{Q_i}(h_1) - v_{Q_i}(h_2))^2 \Big)^{\frac{1}{2}}. \tag{4.2}$$

For $Q \subset \mathbb{R}^d$, similar to $\mathtt{d}_Q$ in $\mathbb{R}^2$, we want to consider the case that there are $d+1$ points in Q which are not on the same hyperplane. We refer to such a point set Q as *full rank* since if we treat the points as rows, and stack them to form a matrix, then that matrix is full rank. Like lines in $\mathbb{R}^2$, a hyperplane can also be mapped to a point in $\mathbb{R}^n$, and if Q is full rank, then no two hyperplanes will be mapped to the same point in $\mathbb{R}^n$. So, similar to Theorem 4.1, we can prove $\mathtt{d}_Q$ is a metric in $\mathcal{H}$.

Theorem 4.2. *If $Q = \{q_1, q_2, \cdots, q_n\} \subset \mathbb{R}^d$ is full rank, then $\mathtt{d}_Q$ is a metric in $\mathcal{H}$.*

Remark 4.2. *The distance can be generalized to weighted point sets and continuous probability distributions. Suppose $Q = \{q_1, \cdots, q_n\} \subset \mathbb{R}^d$, $W = \{w_1, \cdots, w_n\} \subset (0, \infty)$, and μ is a probability measure on $\mathbb{R}^d$. For two hyperplanes h_1, h_2 in $\mathbb{R}^d$, we define*

$$\mathtt{d}_{Q,W}(h_1, h_2) = \Big(\sum_{i=1}^{n} w_i (v_{Q_i}(h_1) - v_{Q_i}(h_2))^2 \Big)^{\frac{1}{2}},$$
$$\mathtt{d}_{\mu}(h_1, h_2) = \Big(\int_{\mathbb{R}^d} (v_x(h_1) - v_x(h_2))^2 \mathtt{d}\mu(x) \Big)^{\frac{1}{2}},$$

where $v_x(\cdot)$ is defined in the same way as $v_{Q_i}(\cdot)$ for $x \in \mathbb{R}^d$.

4.2.3 VC-Dimension of Metric Balls for $\mathtt{d}_Q$

The distance $\mathtt{d}_Q$ can induce a range space $(\mathcal{H}, \mathcal{R}_Q)$, where again $\mathcal{H}$ is the collection of all hyperplanes in $\mathbb{R}^d$, and $\mathcal{R}_Q = \{B_Q(h, r) \mid h \in \mathcal{H}, r \geq 0\}$ with metric ball $B_Q(h, r) = \{h' \in \mathcal{H} \mid \mathtt{d}_Q(h, h') \leq r\}$. We prove that the VC dimension [85] of this range space only depends on d, and is independent of the number of points in Q.

Theorem 4.3. *Suppose $Q \subset \mathbb{R}^d$ is full rank, then the VC-dimension of the range space $(\mathcal{H}, \mathcal{R}_Q)$ is at most $\frac{1}{2}(d^2 + 5d + 6)$.*

Proof. For any $B_Q(h_0, r) \in \mathcal{R}_Q$, suppose $Q = \{x_1, \cdots, x_n\}$ with $x_i = (x_{i,1}, \cdots, x_{i,d})$ and $h \in B_Q(h_0, r)$. This implies $\mathtt{d}_Q(h, h_0) \leq r$, so if h is represented by a unique vector $(u_1, \cdots, u_{d+1}) \in \mathbb{U}^{d+1}$, then we have

$$\sum_{i=1}^{n} \frac{1}{n} \Big(\sum_{j=1}^{d} u_j x_{i,j} + u_{d+1} - v_{Q_i}(h_0) \Big)^2 \leq r^2. \tag{4.3}$$

Since this can be viewed as a polynomial of $u_1, \cdots, u_{d+1}$, we can use a standard lifting map to convert it to a linear equation about new variables, and then use the VC-dimension of the collection of halfspaces to prove the result.

To this end, we introduce the following data parameters a_j [for $0 \leq j \leq d+1$] and $a_{j,j'}$ [for $1 \leq j \leq j' \leq d+1$] which only depend on Q, h_0, and r. That is these only depend on the metric $\mathtt{d}_Q$ and the choice of metric ball.

$$\begin{aligned}
&a_0 = \sum_{i=1}^{n} v_{Q_i}(h_0)^2 - nr^2, \quad a_{d+1} = -2 \sum_{i=1}^{n} v_{Q_i}(h_0), \\
&a_j = -2 \sum_{i=1}^{n} x_{i,j} v_{Q_i}(h_0) \text{ [for } 1 \leq j \leq d], \\
&a_{d+1,d+1} = n, \quad a_{j,d+1} = 2 \sum_{i=1}^{n} x_{i,j} \text{ [for } 1 \leq j \leq d], \\
&a_{j,j} = \sum_{i=1}^{n} x_{i,j}^2 \text{ [for } 1 \leq j \leq d], \quad \text{and} \\
&a_{j,j'} = 2 \sum_{i=1}^{n} x_{i,j} x_{i,j'} \text{[for } 1 \leq j < j' \leq d].
\end{aligned}$$

We also introduce another set of new variables y_j [for $1 \leq j \leq d+1$] and $y_{j,j'}$ [for $1 \leq j \leq j' \leq d+1$] which only depend on the choice of h:

$$y_j = u_j \text{ [for } 1 \leq j \leq d+1] \text{ and } y_{j,j'} = u_j u_{j'} \text{ [for } 1 \leq j \leq j' \leq d+1].$$

Now (4.3) can be further rewritten as

$$\sum_{j=1}^{d+1} a_j y_j + \sum_{1 \leq j \leq j' \leq d+1} a_{j,j'} y_{j,j'} + a_0 \leq 0.$$

Since the a_j and $a_{j,j'}$ only depend on $\mathtt{d}_Q$, h_0, and r, and the above equation holds for any y_j and $y_{j,j'}$ implied by an $h \in B_Q(h_0, r)$, then it converts $B_Q(h_0, r)$ into a halfspace in $\mathbb{R}^{d'}$ where $d' = 2(d+1) + \binom{d+1}{2} = \frac{1}{2}(d^2 + 5d + 4)$. Since the VC-dimension of halfspaces in $\mathbb{R}^{d'}$ is $d' + 1$, the VC dimension of $(\mathcal{H}, \mathcal{R}_Q)$ is at most $d' + 1 = \frac{1}{2}(d^2 + 5d + 6)$. □

Remark 4.3. *This distance, metric property, and VC-dimension result extend to operate between any objects, such as polynomial models of regression, when linearized to hyperplanes in* $\mathbb{R}^d$.

4.2.4 Unsigned Variant for the Distance Between Lines and Hyperplans

There are several other nicely defined variants of this distance. For a line ℓ we could define $\hat{v}_{Q_i}(\ell) = |v_{Q_i}(\ell)|$, as the *unsigned* distance from $q_i \in Q$ to the line ℓ. When we consider the distance from q_i to some bounded object (e.g., a trajectory in place of ℓ), this distance is more natural. We are able to show that under similar mild restrictions on Q that this is a metric; the condition requires 5 points instead of 3. However, we are not able to show constant-size VC-dimension for its metric balls (as we do for $\mathtt{d}_Q$ in Section 4.2.3).

Suppose $Q = \{q_1, q_2, \cdots, q_n\} \subset \mathbb{R}^2$, $\ell_1, \ell_2 \in \mathcal{L} = \{\ell \mid \ell \text{ is a line in } \mathbb{R}^2\}$. Given $\ell \in \mathcal{L}$, we write ℓ in the form as before and define $\hat{v}_Q(\ell) = (\hat{v}_{Q_1}(\ell), \hat{v}_{Q_2}(\ell), \ldots, \hat{v}_{Q_n}(\ell))$ where $\hat{v}_{Q_i}(\ell) = |u_1 x_i + u_2 y_i + u_3|$ and (x_i, y_i) is the coordinates of $q_i \in Q$, and then define the first variant of $\mathtt{d}_Q$ as

$$\hat{\mathtt{d}}_Q(\ell_1, \ell_2) := \Big\| \frac{1}{\sqrt{n}} (\hat{v}_Q(\ell_1) - \hat{v}_Q(\ell_2)) \Big\| = \Big(\sum_{i=1}^{n} \frac{1}{n} (\hat{v}_{Q_i}(\ell_1) - \hat{v}_{Q_i}(\ell_2))^2 \Big)^{\frac{1}{2}}. \tag{4.4}$$

For (4.4), we have the following theorem.

Theorem 4.4. *Suppose in $Q \subset \mathbb{R}^2$ there is a subset of five points, and any three points in this subset are non-collinear, then $\hat{\mathtt{d}}_Q$ is a metric in $\mathcal{L}$.*

Proof. We only need to show if $\hat{\mathtt{d}}_Q(\ell_1, \ell_2) = 0$, then $\ell_1 = \ell_2$. Suppose $\widetilde{Q} = \{q_1, \cdots, q_5\} \subset Q$, and any three points in $\widetilde{Q}$ are not on the same line. If $\ell_1 \neq \ell_2$, then let ℓ_1' and ℓ_2' be the two bisectors of the angles formed by ℓ_1 and ℓ_2. From $\hat{\mathtt{d}}_Q(\ell_1, \ell_2) = 0$, we know $\hat{v}_{Q_i}(\ell_1) = \hat{v}_{Q_i}(\ell_2)$ for $i \in [5]$, which means the distances from $q_i \in \widetilde{Q}$ to ℓ_1 and to ℓ_2 are equal. So, any point $q_i \in \widetilde{Q}$ must be either on ℓ_1' or on ℓ_2', which implies there must be three collinear points in $\widetilde{Q}$. This is contradictory to the fact that any three points in $\widetilde{Q}$ are not on the same line. $\square$

Remark 4.4. *Definition* (4.4) *can be generalized to hyperplanes in $\mathbb{R}^d$:*

$$\hat{\mathtt{d}}_Q(h_1, h_2) := \big(\sum_{i=1}^{n} \frac{1}{n} (\hat{v}_{Q_i}(h_1) - \hat{v}_{Q_i}(h_2))^2 \big)^{\frac{1}{2}}, \tag{4.5}$$

where $h_1, h_2 \in \mathcal{H}$, and $\hat{v}_{Q_i}(h_j)$ is the distance from point q_i in $Q \subset \mathbb{R}^d$ to h_j ($j = 1, 2$). Using the similar method, we can show if there is a subset of $2d + 1$ points in Q and any $d + 1$ points in this subset are not on the same hyperplane, then (4.5) *is a metric in $\mathcal{H}$.*

Matrix Norm Variant. In another variant of $\mathtt{d}_Q$ we define $\tilde{v}_{Q_i}(\ell)$ as a *vector* from q_i to the closest point on ℓ. More specifically, suppose ℓ is in the same form as before, then the projection of point $q_i = (x_i, y_i)$ on ℓ is $(\tilde{x}_i, \tilde{y}_i) = (x_i \cos^2(\alpha) - y_i \sin(\alpha)\cos(\alpha) - c\sin(\alpha), -x_i \cos(\alpha)\sin(\alpha) + y_i \sin^2(\alpha) - c\cos(\alpha))$, and we define $\tilde{v}_{Q_i}(\ell) = (\tilde{x}_i - x_i, \tilde{y}_i - y_i)$ for $(x_i, y_i) \in Q$, and an $n \times 2$ matrix $V_{Q,l} = [\tilde{v}_{Q_1}(\ell); \cdots ; \tilde{v}_{Q_n}(\ell)]$ where $\tilde{v}_{Q_i}(\ell)$ is the ith row of $V_{Q,l}$. For $\ell_1, \ell_2 \in \mathcal{L}$ we define the distance between these two lines as

$$\tilde{\mathtt{d}}_Q(\ell_1, \ell_2) := \|V_{Q,\ell_1} - V_{Q,\ell_2}\|_F, \tag{4.6}$$

where $\|\cdot\|_F$ is the Frobenius norm of matrices. For (4.6), we have the following theorem.

Theorem 4.5. *Suppose in $Q \subset \mathbb{R}^2$ there are two different points q_1 and q_2, then $\tilde{\mathtt{d}}_Q$ is a metric in $\mathcal{L}$.*

Proof. We only need to show if $\tilde{\mathtt{d}}_Q(\ell_1, \ell_2) = 0$, then $\ell_1 = \ell_2$. There are two cases.

(1) $\tilde{v}_{Q_1}(\ell_1) = (0,0)$ and $\tilde{v}_{Q_2}(\ell_1) = (0,0)$. From $\tilde{\mathtt{d}}_Q(\ell_1, \ell_2) = 0$ we know $\tilde{v}_{Q_1}(\ell_2) = 0$ and $\tilde{v}_{Q_2}(\ell_2) = 0$, which means q_1 and q_2 are on both ℓ_1 and ℓ_2, so $\ell_1 = \ell_2$.

(2) $\tilde{v}_{Q_1}(\ell_1) \neq (0,0)$ or $\tilde{v}_{Q_2}(\ell_1) \neq (0,0)$. In this case, without loss of generality we assume $\tilde{v}_{Q_1}(\ell_1) \neq (0,0)$. From $\tilde{\mathtt{d}}_Q(\ell_1, \ell_2) = 0$ we have $\tilde{v}_{Q_1}(\ell_2) = \tilde{v}_{Q_1}(\ell_1) \neq (0,0)$, so introducing the notation $(\tilde{x}_i - x_i, \tilde{y}_i - y_i) = \tilde{v}_{Q_1}(\ell_1)$, we know $(\tilde{x}_i, \tilde{y}_i)$ is on ℓ_1 and ℓ_2, and $\tilde{v}_{Q_1}(\ell_1)$ is the normal direction of ℓ_1 and ℓ_2. Since a point and a normal direction can uniquely determine a line, we have $\ell_1 = \ell_2$. □

Remark 4.5. *Definition* (4.6) *can be generalized to hyperplanes in $\mathbb{R}^d$:*

$$\tilde{\mathtt{d}}_Q(h_1, h_2) := \|V_{Q,h_1} - V_{Q,h_2}\|_F, \tag{4.7}$$

where $h_1, h_2 \in \mathcal{H}$, and V_{Q,h_j} $(j = 1,2)$ is an $n \times d$ matrix with each row being a projection vector from a point in Q to h_j. Using the similar method, we can show if there are d different points in Q, then (4.7) *is a metric in $\mathcal{H}$.*

4.2.5 Applications in Analysis

The new distance $\mathtt{d}_Q$ for hyperplanes has many applications in statistical and algorithmic data analysis where hyperplanes map to linear models. For example, the vectorized representation implies we can use Lloyd's algorithm for k-means clustering on lines or

hyperplanes, and the metric property implies Gonzalez algorithm [49] for k-center clustering will give a 2-approximation. Here we elaborate some algorithm applications of $\mathtt{d}_Q$ and its stability.

Kernel Density Estimates. Given a large varieties of regression models $H = \{h_1, h_2, \ldots, h_m\}$ (e.g., stemming from different algorithms or model parameters) we can define a Gaussian-type kernel $K(h_1, h_2) = \exp(-\mathtt{d}_Q(h_1, h_2)^2)/Z$ using $\mathtt{d}_Q$ as the underlying metric, and with an appropriate normalization constant Z. Then for any regressor h, the kernel density estimate is defined $\textsc{kde}_H(h) = \frac{1}{|H|}\sum_{h_i \in H} K(h, h_i)$.

The constant VC-dimension of the metric balls of $\mathtt{d}_Q$ from Theorem 4.3 indicates that despite the complex nature of this distance and high-dimensional embedding, this may indeed be feasible. For instance, Joshi *et.al.*[61] considered kernels where the range space defined by superlevel sets of any kernel have bounded VC-dimension ν. Then for a data set X, a random sample $Y \subset X$ of size $O(\frac{1}{\varepsilon^2}(\nu + \log\frac{1}{\delta}))$ approximates the $\textsc{kde}_X$ at any evaluation point so that $|\textsc{kde}_X(x) - \textsc{kde}_Y(x)| \leq \varepsilon$, with probability at least $1 - \delta$. In the case of our $\mathtt{d}_Q$ based kernels, by Theorem 4.3 it indicates that a random sample $J \subset H$ of size $O(\frac{1}{\varepsilon^2}(d^2 + \log\frac{1}{\delta})$ (with normalization factor $Z = 1$) is sufficient so that with probability at least $1 - \delta$, then for any evaluation regressor h that $|\textsc{kde}_H(h) - \textsc{kde}_J(h)| \leq \varepsilon$. Alternatively, if H represents the set of *all* possible bootstrapped samples, then we only need to generate $m = O(\frac{1}{\varepsilon^2}(d^2 + \log\frac{1}{\delta}))$ point sets and hyperplanes J to get a ε-approximate estimate of this density. Then we can run a mode detection algorithm [21] to determine the modality.

Approximating the Siegel Estimator Distribution on Uncertain Data. The Siegel estimator [81] as discussed in Section 2.4.2 is an example of a robust estimator S for linear regression; given a set $P = \{p_1, p_2, \cdots, p_n\} \subset \mathbb{R}^2$, it returns a line $S(P) : y = ax + b$ to fit these n points.

Now consider a set of n uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$, where the ith point is represented by a discrete set of k possible locations $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}^2$. Define $P_{\mathsf{flat}} = \cup_{i=1}^n P_i$, and say $A \Subset \mathcal{P}$ is a traversal of $\mathcal{P}$ if $A = \{a_1, \ldots a_n\}$ has each a_i in the domain of P_i. A robust way to understand the uncertainty of the data [67] is to build a distribution over the outcomes $S(A)$ for $A \Subset \mathcal{P}$. To do this, we apply Algorithm 2.2, which can be rewritten as Algorithm 4.1.

Algorithm 4.1 Approximate Siegel estimator on uncertain data

Initialize $T = \emptyset$, the set of possible Siegel estimators.
for $j = 1$ to N **do**
  Randomly choose $A \Subset \mathcal{P}$; add $z = S(A)$ to T.
return Multiset T

Suppose $\mathcal{L} = \{\ell \mid \ell \text{ is a line in } \mathbb{R}^2\}$, and we use the metric $\mathtt{d}_Q(\cdot,\cdot)$ in $\mathcal{L}$, with $Q = P_{\mathsf{flat}}$. From Theorem 4.3 the VC dimension of $(\mathcal{L}, \mathcal{R}_{P_{\mathsf{flat}}})$ is a constant. Therefore, as corollary of Theorem 2.9, we have the following result.

Corollary 4.1. *For error parameters $\varepsilon > 0$ and $\delta \in (0,1)$, run Algorithm 4.1 with $N = O((1/\varepsilon^2)(\log(1/\delta)))$, to obtain multiset T. Then, with probability at least $1-\delta$, for any $z \in \mathcal{L}$ and radius r we have $||T \cap B_Q(z,r)|/N - \Pr_{A \Subset \mathcal{P}}[S(A) \in B_Q(z,r)]| \leq \varepsilon$.*

Multi-Modality Detection. There are many scenarios in which one may generate a large set of possible regressions. One may run various algorithms, or use many parameters for one algorithm, each generating a separate regression. Or to understand the variance inherent in the data, bootstrapping is a common technique. In this setting, from a data set $Q \subset \mathbb{R}^d$ of size n, one randomly samples m data sets $X_1, X_2, \ldots, X_m$, each of size n from Q with replacement. Then for each data set X_i, we run a regression formulation to generate a hyperplane h_i. This induces a set $H = \{h_1, \ldots, h_m\}$ of hyperplanes.

Since the distance between two hyperplanes has been defined and is a metric, we can run k-center clustering (for instance with Gonzalez algorithm [49]) on the set $\{h_1, \cdots, h_m\}$. Then we use "elbow method" to find the appropriate value of k: if the cost (in this case the largest distance from some h_i to the representative center regressor) drops dramatically up until the kth center is found, and then it levels off as more centers are added, it implies there are probably k natural clusters. If the appropriate value of k is greater than 1, it implies multi-modality in Q with respect to linear models.

For example, in Figure 4.2, suppose Q is a set of n points in $\mathbb{R}^2$, and some points in Q are around the line ℓ, but others are around ℓ'. For a set of bootstrapped samples $X_1, X_2, \ldots$ from Q, we would expect some robust regression algorithms would fit ℓ_i to X_i so ℓ_i is close to ℓ, and for other ℓ_j fit to X_j so that ℓ_j would be closer to ℓ'. Then likely running the elbow technique on this set of $\{\ell_i\}_{i \in [1,m]}$ would result in an estimate of $k = 2$, indicating

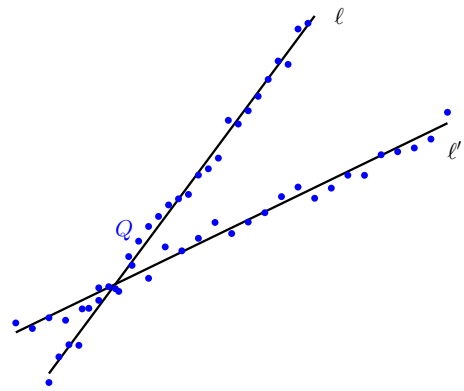

Figure 4.2: Multi-modality in regression.

multi-modality.

In this process, if Q is very large, we can use the methods in Section 5.2.1 to compute $\mathtt{d}_Q$ approximately, and the clustering should still be accurate enough to distinguish multi-modality (Gonzalez only provides a 2-approximation regardless).

Coreset Evaluation in Regression. Given a finite point set $Q \subset \mathbb{R}^d$, the linear regression problem can usually be formalized as: finding $h \in \mathcal{H} = \{h \mid h \text{ is a hyperplane in } \mathbb{R}^d\}$ to minimize $\text{cost}(Q, h)$, where the cost function depends on different regression models.

Using a coreset $\hat{Q}$ as a replacement of Q can simplify the computation when Q is very large. A ε-coreset [41, 42] for Q is a set $\hat{Q} \subset \mathbb{R}^d$ such that $(1-\varepsilon)\text{cost}(Q, h) \leq \text{cost}(\hat{Q}, h) \leq (1+\varepsilon)\text{cost}(Q, h)$, for all $h \in \mathcal{H}$. Suppose $h^* = \arg\min_{h \in \mathcal{H}} \text{cost}(Q, h)$, and $\hat{h} = \arg\min_{h \in \mathcal{H}} \text{cost}(\hat{Q}, h)$. We can use $\text{cost}(Q, \hat{h}) - \text{cost}(Q, h^*)$ to evaluate how well the coreset $\hat{Q}$ approximates Q.

However, suppose $\hat{Q}_1$ and $\hat{Q}_2$ are two ε-coresets of Q, and $h_1 = \arg\min_{h \in \mathcal{H}} \text{cost}(\hat{Q}_1, h)$, $h_2 = \arg\min_{h \in \mathcal{H}} \text{cost}(\hat{Q}_2, h)$. If $\text{cost}(Q, h_1) - \text{cost}(Q, h^*)$ and $\text{cost}(Q, h_2) - \text{cost}(Q, h^*)$ are very close to each other, which one is a better coreset, $\hat{Q}_1$ or $\hat{Q}_2$? In this situation, we can directly compare $\mathtt{d}_Q(h^*, h_1)$ and $\mathtt{d}_Q(h^*, h_2)$. If $\mathtt{d}_Q(h^*, h_1) < \mathtt{d}_Q(h^*, h_2)$, then it means $\hat{Q}_1$ does a better job of inducing a model h_1 that fits the full data more similarly to how h^* would than h_2 resulting from $\hat{Q}_2$. This provides a more fine-grained method to evaluate coresets; it describes not just how well they fit the data, but also how similar it is to how an optimal classifier would fit the same data. With other distance measures, this sort of analysis was not available.

4.2.6 Direct Extension (literally) to Trajectories

In this section, we show how $\mathtt{d}_Q$ can be simply generalized to the distance between two piecewise-linear curves, while retaining the many nice properties described above. Let $\mathcal{T}_k = \{\gamma \mid \gamma \text{ is a curve in } \mathbb{R}^2 \text{ defined by } k \text{ ordered line segments}\}$ represent the space of all k-piecewise linear curves.

For any curve $\gamma \in \mathcal{T}_k$, let its k segments be $\langle s_1, s_2, \ldots, s_k \rangle$, and let these map to k lines $\ell_1, \ldots, \ell_k$ where each ℓ_j contains s_j (literally an "extension" of s_j to a line ℓ_j). Next add two more lines: ℓ_0 which is perpendicular to ℓ_1 and passes through the first end point of s_1, and ℓ_{k+1} which is perpendicular to ℓ_k and passes through the last end point of s_k (in high dimensions, some canonical choice of ℓ_0 and ℓ_{k+1} is needed). We now represent γ as the ordered set of $k+2$ lines $(\ell_0, \ell_1, \ldots, \ell_k, \ell_{k+1})$. This mapping is 1 to 1, since segments s_i and s_{i+1} share a common end point, and this defines the intersection between ℓ_i and ℓ_{i+1}. The intersections with the added lines ℓ_0 and ℓ_{k+1} define first and last endpoints of s_1 and s_k, and these endpoints are sufficient to define γ.

Now for two curves $\gamma^{(1)}, \gamma^{(2)} \in \Gamma_k$, we define the distance using their line representations $(\ell_0^{(1)}, \ldots, \ell_{k+1}^{(1)})$ and $(\ell_0^{(2)}, \ldots, \ell_{k+1}^{(2)})$, respectively, as

$$\mathtt{d}_Q^{\leftrightarrow}(\gamma^{(1)}, \gamma^{(2)}) := \frac{1}{k+2}\Big(\textstyle\sum_{i=0}^{k+1} \mathtt{d}_Q\big(\ell_i^{(1)}, \ell_i^{(2)}\big)\Big).$$

Metric. If $\mathtt{d}_Q^{\leftrightarrow}(\gamma^{(1)}, \gamma^{(2)}) = 0$, then $\mathtt{d}_Q\big(\ell_i^{(1)}, \ell_i^{(2)}\big) = 0$ for all $i \in [k]$, which implies $\ell_i^{(1)} = \ell_i^{(2)}$ if Q is full rank. Combined with the 1to1 nature of the mapping from $\gamma = (s_1, \ldots, s_k)$ to $(\ell_0, \ldots, \ell_{k+1})$, we have that if Q is full rank, then $\mathtt{d}_Q^{\leftrightarrow}$ is a metric over $\mathcal{T}_k$.

VC dimension. The distance $\mathtt{d}_Q^{\leftrightarrow}(\cdot,\cdot)$ can induce a range space $(\mathcal{T}_k, \mathcal{S}_{Q,k})$, where again $\mathcal{T}_k$ is the collection of all k-piecewise linear curves in $\mathbb{R}^2$, and $\mathcal{S}_{Q,k} = \{B_Q(\gamma, r) \mid \gamma \in \mathcal{T}_k, r \geq 0\}$ with metric ball $B_Q(\gamma, r) = \{\gamma' \in \mathcal{T}_k \mid \mathtt{d}_Q^{\leftrightarrow}(\gamma, \gamma') \leq r\}$. Using the straightforward extensions of the method in the proof of Theorem 4.3, we can show the VC dimension of this range space only depends on k, and is independent of the number of points in Q. Specifically, for full rank $Q \subset \mathbb{R}^2$, the VC-dimension of $(\mathcal{T}_k, \mathcal{S}_{Q,k})$ is at most $9k + 19$.

While retaining all above mathematical properties, this distance is unintuitive, and as we show in Section 4.4, can perform less than optimally. We next develop other trajectory distances which are more intuitive, but have weaker mathematical properties.

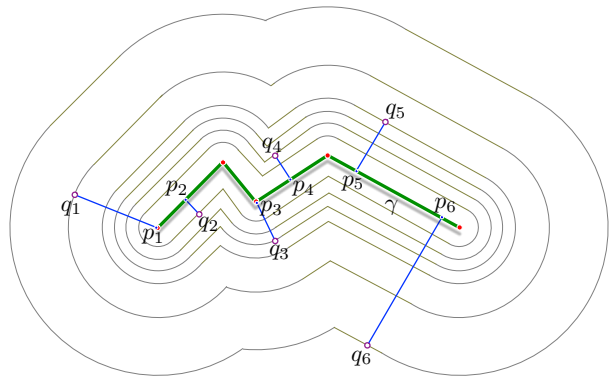

Figure 4.3: Illustrating q_i and p_i on a trajectory for $\mathtt{d}_Q$ and $\mathtt{d}_Q^\pi$.

4.3 Landmark Distances Between Trajectories

In this section, we define two variants of $\mathtt{d}_Q$ for trajectories, focused on their modeling as piecewise-linear curves on $\mathbb{R}^2$. We let $\mathcal{T}$ define the set of such curves, and they are specified by a series of critical points $\langle c_0, c_1, \ldots, c_k \rangle$. The curve $\gamma \in \mathcal{T}$ is the subset of $\mathbb{R}^2$ defined by the k segments $s_1, s_2, \ldots, s_k$ where $s_i = \overline{c_{i-1}c_i}$ is the continuous set of points between critical points c_{i-1} and c_i. For notational convenience, we will describe all curves as having k segments, but the distance will not require this. Moreover, since we model the trajectory as a continuous subset of $\mathbb{R}^2$, it will not distinguish trajectories of different speeds or moving in opposite directions but following the same paths.

Now for a curve $\gamma \in \mathcal{T}$ and size n point set $Q \subset \mathbb{R}^2$, define $v_i = \min_{p \in \gamma} \|q_i - p\|$ and $p_i = \arg\min_{p \in \gamma} \|q_i - p\|$; see Figure 4.3. If $\arg\min_{p \in \gamma} \|q_i - p\|$ is not unique, then we take the point with smallest x-cordinate (or smallest y-coordinate when more than two points have the same smallest x-coordinate) as p_i. For two curves $\gamma^{(1)}$ and $\gamma^{(2)}$ denote these values as $v_i^{(1)}$, $p_i^{(1)}$ and $v_i^{(2)}$, $p_i^{(2)}$ respectively. Our distances are then defined as:

$$\mathtt{d}_Q(\gamma^{(1)}, \gamma^{(2)}) = \Big(\frac{1}{n}\sum_{i=1}^{n}\big(v_i^{(1)} - v_i^{(2)}\big)^2\Big)^{\frac{1}{2}}, \;\; \mathtt{d}_Q^\pi(\gamma^{(1)}, \gamma^{(2)}) = \frac{1}{n}\sum_{i=1}^{n}\big(\|p_i^{(1)} - p_i^{(2)}\|\big).$$

The standard variant $\mathtt{d}_Q$ is the analog of the version for halfspaces, where as the second variant $\mathtt{d}_Q^\pi$ (the *projected landmark distance*) projects Q onto the closest points of the curves, and then computes the average distances with respect to these projected points.

4.3.1 Metric Properties

In this section, we show a reasonable condition for the trajectories and Q so that both variants are metrics. As with lines and halfspaces, these distances are always pseudometrics: the symmetry and triangle inequality are direct consequences of the embedding to Euclidean space. The only restriction of the trajectories is to ensure that two distinct curves do not have a distance 0, and in our arguments this requires that the critical points have some non-zero separation from other parts of the curve. These restrictions may not be necessary, but it makes the proofs simple enough. Then we basically just require that Q is sufficiently dense; if we decide many of these points are irrelevant, we can reduce the weights on those points (keeping them non-zero) and the metric properties still hold.

We define a family of curves $\mathcal{T}_\tau \subset \mathcal{T}$ so each $\gamma \in \mathcal{T}_\tau$ has two restrictions: (R1) Each angle $\angle_{[c_{i-1},c_i,c_{i+1}]}$ about an internal critical point c_i is non-zero (i.e., in $(0,\pi)$). (R2) Each critical point c_i is τ-*separated*, that is the ball $B(c_i,\tau) = \{x \in \mathbb{R}^2 \mid \|x - c_i\| \leq \tau\}$ only intersects the two adjacent segments s_{i-1} and s_i of γ, or one adjacent segment for end points (i.e., only the s_1 for c_0 and s_k for c_k, if γ has k line segments). The τ-separated property, for instance, enforces that critical points are at least a distance τ apart.

We next restrict that all curves (and Q) lie in a sufficiently large bounded region $\Omega \subset \mathbb{R}^2$. Let $\mathcal{T}_\tau(\Omega)$ be the subset of $\mathcal{T}_\tau$ where all curves γ have all critical points within Ω, and in particular, no $c_i \in \gamma \in \mathcal{T}_\tau(\Omega)$ is within a distance τ of the boundary of Ω. Now for $\eta > 0$, define an infinite grid $G_\eta = \{g_v \in \mathbb{R}^2 |\ g_v = \eta v \text{ for } v = (v_1, v_2) \in \mathbb{Z}^2\}$, where $\mathbb{Z}$ is all integers.

Theorem 4.6. *For $Q = G_\eta \cap \Omega$ and $\eta \leq \frac{\tau}{16}$, both $\mathtt{d}_Q$ and $\mathtt{d}_Q^\pi$ are metrics in $\mathcal{T}_\tau(\Omega)$.*

Proof. We prove this theorem for $\mathtt{d}_Q^\pi$, and the proof for $\mathtt{d}_Q$ is similar and given in Appendix B.1. Suppose $\gamma^{(1)}, \gamma^{(2)} \in \mathcal{T}_\tau(\Omega)$ have critical points $c_0, c_1, ...c_k$ and $c'_0, c'_1, ...c'_{k'}$ respectively. We only need to show if $\mathtt{d}_Q^\pi(\gamma^{(1)}, \gamma^{(2)}) = 0$ then $\gamma^{(1)} = \gamma^{(2)}$. Here, if two piecewise-linear curves have the same critical points and their orders are the same or reverse of each other, then these two curves are regarded as the same curve.

The argument follows 4 steps assuming $\mathtt{d}_Q^\pi(\gamma^{(1)}, \gamma^{(2)}) = 0$: (Step 1) Around each critical point c_i of $\gamma^{(1)}$, we can identify at least 4 points q_1, q_2, q_3, q_4 that map to p_1, p_2, p_3, p_4, two each on the two segments adjacent to c_i. (Step 2) The segments between defined by $\overline{p_1 p_2}$

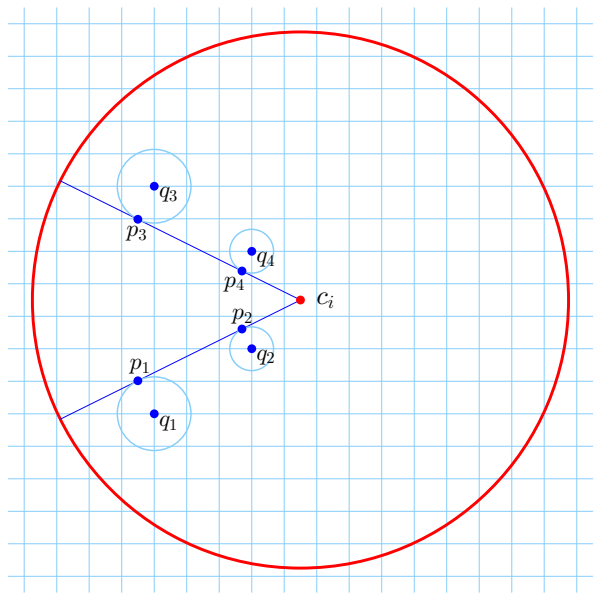

Figure 4.4: c_i is a critical point of $\gamma^{(1)}$

and $\overline{p_3p_4}$ must also be part of $\gamma^{(2)}$. (Step 3) The line extension of those two line segment must intersect at c_i, and this must also be critical point on $\gamma^{(2)}$ (Step 4) Because these Steps 1-3 can be repeated for all critical points on $\gamma(1)$ and on $\gamma^{(2)}$, they must share critical points and connecting line segments, and be the same curves.

We formalize these steps based on three observations: (O1) If $\gamma \in \mathcal{T}_\tau$, then in any ball with radius $\frac{\tau}{2}$, there is at most one critical point of γ. (O2) If a point moves along $\gamma \in \mathcal{T}$, then it can only stop or change direction at critical points. (O3) For $q \in Q, \gamma \in \mathcal{T}$, $p = \arg\min_{p' \in \gamma} \|p' - q\|$, suppose l is the tangent line of the circle $C(q, \|q - p\|)$ where q is center and $\|q - p\|$ its radius, at point p. If $l \cap B(p, \delta)$ is not apart of γ for all $\delta > 0$, then p must be a critical point of γ.

<u>Step 1:</u> Suppose $c_i = (x_i, y_i)$ $(1 \leq i \leq k-1)$ is a critical point of $\gamma^{(1)}$, and consider a ball $B(c_i, \frac{1}{2}\tau)$, as shown in Figure 4.4. Since the side length of each grid cell is $\eta \leq \frac{1}{16}\tau$, from the τ-separated property (R2) we know for any $q \in Q \cap B(c_i, \frac{\tau}{2})$, $p = \arg\min_{p' \in \gamma^{(1)}} \|p' - q\|$ is in $B(c_i, \frac{\tau}{2})$. So, there exist two points q_1, q_2 that are mapped to points p_1, p_2 on one line segment of $\gamma^{(1)}$ and another two points q_3, q_4 are mapped to points p_3, p_4 on the other line segment of $\gamma^{(1)}$ in $B(c_i, \frac{\tau}{2})$. Since $\mathrm{d}_Q^\pi(\gamma^{(1)}, \gamma^{(2)}) = 0$, we know p_1, p_2, p_3, p_4 are also on $\gamma^{(2)}$.

<u>Step 2:</u> We assert the line segment $\overline{p_1p_2}$ must be a part of $\gamma^{(2)}$. From (O1), we know p_1 and p_2 cannot both be the critical point of $\gamma^{(2)}$ at the same time, so we assume p_1 is not a critical

point. Thus, from (O3) we know a small part of tangent line l of circle $C(q_1, \|q_1 - p_1\|)$ at p_1 is a part of $\gamma^{(2)}$. If p_2 is a critical point of $\gamma^{(2)}$, then from (O1) and (O2) we know the line segment $\overline{p_1 p_2}$ must be a part of $\gamma^{(2)}$. If p_2 is not a critical point of $\gamma^{(2)}$, then from (O3) we know a small part of tangent line l of circle $C(q_1, \|q_1 - p_1\|)$ at p_2 is a part of $\gamma^{(2)}$. So, in this case, (O1) and (O2) implies the line segment $\overline{p_1 p_2}$ is a part of $\gamma^{(2)}$. Using a similar argument, we know the line segment $\overline{p_3 p_4}$ is also a part of $\gamma^{(2)}$.

Step 3: We extend the line $\overline{p_1 p_2}$ from p_1 to p_2 and the line $\overline{p_3 p_4}$ from p_3 to p_4. Suppose they intersect with the boundary of $B(c_i, \frac{\tau}{2})$ at p_2' and p_4' respectively. Since $\gamma^{(2)}$ cannot go into the interior of any ball with centers in $Q \cap B(c_i, \frac{\tau}{2})$, from (O2) we know there must be one critical point in line segment $\overline{p_2 p_2'}$. For the same reason, there must be one critical point in line segment $\overline{p_4 p_4'}$. Thus, (O1) implies c_i is a critical points of $\gamma^{(2)}$.

Step 4: Considering that $\gamma^{(2)}$ has to pass through p_1, p_2, p_3, p_4 and c_i, from τ-separated property (R2), we know $\gamma^{(1)}$ and $\gamma^{(2)}$ must overlap with each other in $B(c_i, \frac{\tau}{2})$. For two endpoints c_0 and c_k we can make the same argument, which means in a neighborhood of each critical point of $\gamma^{(1)}$, $\gamma^{(1)}$ overlaps with $\gamma^{(2)}$. This means $\{c_0, c_1, \cdots, c_k\}$ is a subset of $\{c_0', c_1', \cdots, c_{k'}'\}$. Using the same argument $\{c_0', c_1', \cdots, c_{k'}'\}$ is a subset of $\{c_0, c_1, \cdots, c_k\}$. Therefore, $k = k'$ and we know $\gamma^{(1)}$ and $\gamma^{(2)}$ must have the same critical points and their orders must be the same or reverse of each other. □

Remark 4.6. *We did not try to optimize constants. The point is that for most families of trajectories, with Q sufficiently dense our distances are metrics, not just pseudometrics. In practice these distances will work for small sets Q (see below).*

4.4 Trajectories Analysis via New Distances

We demonstrate that $\mathtt{d}_Q$ and $\mathtt{d}_Q^{\pi}$ (and to lesser extent $\mathtt{d}_Q^{\leftrightarrow}$) work effectively on real world problems. These approaches achieve state-of-the-art performance, are incredibly simple to use, and their sketched representation plugs directly into k-means clustering, KNN or SVM classifiers, or ANN libraries. We show that only a small number of landmarks are needed for good accuracy, and when certain landmarks are especially meaningful, our approaches can be easily tuned to achieve very high accuracy.

4.4.1 Related Trajectory Distances, and Landmarks

There are by now numerous definitions of trajectories, with a variety of different aspects they can model and take into account.

We compare the classification errors found using $\mathtt{d}_Q^{\leftrightarrow}$, $\mathtt{d}_Q$ and $\mathtt{d}_Q^{\pi}$ with a series of representative distances for trajectories. These are: Euclidean distance among the critical points (Eu) [95], discrete Frechet distance (dF) [38], dynamic time warping distance (DTW) [93], discrete Hausdorff distance (dH) [71], longest common subsequence distance (LCSS) [87], edit distance for real sequences (EDR) [23]. We also compare against the recently proposed locality sensitive hashing distance (LSH1$_Q$), and the ordered version of locality sensitive hashing distance (LSH2$_Q$) [13], which consider the intersection of the trajectories with a set of disks. This is conceptually similar to our methods, where we can think of the landmarks Q as the centers of disks (as we do in experiments), and their approach requires a radius parameter r for all disks, and is not a metric. The definitions of these distances are given in Appendix B.2.

To find the best parameters to minimized the error, for LCSS we tested $\varepsilon \in \{0.001, 0.005, 0.01, 0.015, \cdots, 0.055\}$, $\delta \in \{1, 2, 3, \cdots, 10\}$, and for EDR we tested $\varepsilon \in \{0.001, 0.005, 0.01, 0.015, \cdots, 0.055\}$, and for LSH1$_Q$ and LSH2$_Q$ we tested $r \in \{0.005, 0.01, 0.02, \cdots, 0.11\}$. Since in all experiments (except Section 4.4.6), each trajectory is represented by a sequence of 10 critical points, it is enough to take the largest value of δ as 10 for LCSS. We only show the best results in this section, but provide the results of other parameter settings in Appendix B.3.

Zhang *et.al.* [95] conducted a large comparison of trajectory distances and showed that in most cases Eu is general enough, efficient, and a superior or nearly as good model as any other ; we include dF and DTW as examples which search over all possible alignments and thus do not require the same number of or aligned critical points on both curves. The restriction that trajectories have the same number of critical points is also not required for dH, EDR, LSH1$_Q$, and LSH2$_Q$, but in comparisons we always first reduce all trajectories to 10 critical points (with Douglas-Peucker), except in Section 4.4.6, so a fair comparison to all metrics can be made. In Appendix B.4, we give the results of reducing all trajectories to at most 40 critical points for Beijing drivers experiment of Section 4.4.3. We do this to see if there is a large effect from trajectory simplification. In general, there is no large effect.

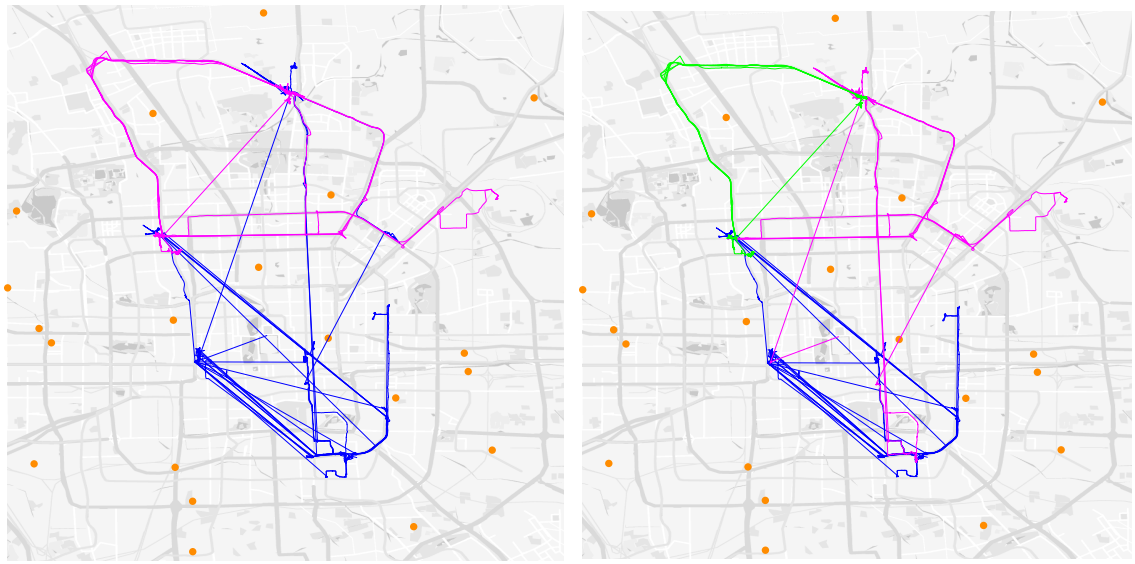

Figure 4.5: 2 or 3 clusters (color-coded) under k-means on $\mathtt{d}_{Q_1}$ with 20 landmarks Q_1 shown overlaid on Beijing.

The performances of most distances are slightly improved, except LCSS. The mean error of LCSS is improved about 8.8%, but to find the best pair of parameters for LCSS needs a lot of computation. More details are discussed in Appendix B.4.

Even beyond the recent trajectory LSH paper [13], the use of waypoints to provide a distance between trajectories is not new. However, they are typically used in other contexts, such as annotating with geolocated social media [91]. Or for instance, in the context of a line of work [46, 50, 66] seeking to find the k nearest time-encoded trajectories to a given point at a specific time, Lin *et.al.* [66] use a set of landmarks Q to map trajectories and query points into the Voronoi cells of Q to quickly help in pruning.

4.4.2 Warm-up: k-means Clustering

As a warm up, we consider clustering the 42 trajectories from user id_{155} in the Geolife GPS trajectory dataset [96]. We randomly choose 20 spread-out Beijing POIs as the landmark set Q_1, shown as orange dots in Figure 4.5. Using $\mathtt{d}_{Q_1}$, this maps each trajectory γ to $\mathbb{R}^{20}$, and we directly run Lloyd's algorithm for k-means clustering with $k = 2, 3$, and color-code the corresponding trajectories in Figure 4.5. We observe that although the trajectories are intertwined, there is a central-city cluster found in both cases, and either 1 or 2 clusters found on the north side.

4.4.3 Classifying Trajectories 1: Beijing Drivers

We also consider classifying trajectories from users in the Geolife dataset [96] with the same 20 POI landmarks Q_1 as in the clustering example. There are 182 users, and each user has several trajectories in Beijing. We only consider those trajectories with more than 10 critical points, and if a user has less than 10 such trajectories, then we remove this user. Thus, 54 users are removed, and in the remaining 128 users, 20 of them have more than 200 trajectories. For each of these users, we just randomly sample 200 trajectories (without replacement), to avoid severe imbalance in classification – dealing with the imbalance challenge is not the focus of this work.

Suppose two users with id_1 and id_2 have two sets of trajectories $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ respectively. Letting $|\mathcal{T}^{(1)}| = m_1$ and $|\mathcal{T}^{(2)}| = m_2$, we randomly sample $\lfloor \frac{3m_1}{10} \rfloor$ trajectories from $\mathcal{T}^{(1)}$ and $\lfloor \frac{3m_2}{10} \rfloor$ trajectories from $\mathcal{T}^{(2)}$ respectively to form a test set, and use the other trajectories in $\mathcal{T}^{(1)} \cup \mathcal{T}^{(2)}$ as the training data. Then we choose an algorithm and metric to do classification, and compute the error. For users with id_1 and id_2, we do this 10 times and take the mean error as $\text{error}(id_1, id_2)$. We compute $\text{error}(id_1, id_2)$ for all 8128 pairs of 128 uses, and then output the mean, median, and standard deviation (SD) of these 8128 errors.

For all of these 10 distances, we use the KNN classification ($K = 5$); see Table 4.1. The lowest error rates of about 7% error is achieved by $\mathtt{d}^{\pi}_{Q_1}$, DTW and LCSS. Then $\mathtt{d}_Q$, Eu, and EDR achieve error about 8%. Other metrics perform worse with for example, dF at 10%, LSH1_{Q_1} at 13%, $\mathtt{d}^{\leftrightarrow}_{Q_1}$ at 17%, and LSH2_{Q_1} at 24% error. Using the standard deviation, by Chebyshev's inequality, over 8128 pairs, these tiers are significant on this data set.

For $\mathtt{d}_{Q_1}$, $\mathtt{d}^{\pi}_{Q_1}$ and Eu, since they map a trajectory to a vector in Euclidean space, we can also directly use SVM to classify these vectors. We use fitcsvm in matlab R2018b and set 'IterationLimit' (the maximum iteration number) as 200,000 for all kernel functions, and set 'KernelScale' as 'auto' for Gaussian kernel. From Table 4.2, we can see for SVM with three kinds of kernel functions, both $\mathtt{d}_{Q_1}$ and $\mathtt{d}^{\pi}_{Q_1}$ are better than Eu. In the case of Gaussian SVM, both $\mathtt{d}_{Q_1}$ and $\mathtt{d}^{\pi}_{Q_1}$ achieve an error rate of about 7% which is less than the about 8% achieved by Eu. Again in this SVM setting $\mathtt{d}^{\leftrightarrow}_{Q_1}$ performs much worse (for Gaussian kernels) or comparable to other measures, about the same as Eu, (linear quadratic kernels).

As we increase the size of Q to 200 (chosen at random), then both $\mathtt{d}_{\tilde{Q}_1}$ and $\mathtt{d}^{\pi}_{\tilde{Q}_1}$ slightly improve in performance, but not drastically, and $\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_1}$ performs about the same. The error

Table 4.1: Classification error on Beijing Drivers with KNN.

	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	$\mathtt{d}_{Q_1}$	$\mathtt{d}^{\pi}_{Q_1}$	Eu	dF	DTW	dH	LCSS	EDR	LSH1$_{Q_1}$	LSH2$_{Q_2}$
best param	-	-	-	-	-	-	-	ε=0.005,δ=10	ε=0.005	r=0.06	r=0.1
mean	0.1703	0.0817	0.0724	0.0811	0.1045	0.0722	0.0883	0.0714	0.0802	0.1290	0.2409
median	0.1458	0.0667	0.0581	0.0654	0.0873	0.0571	0.0722	0.0500	0.0554	0.0949	0.2182
SD	0.1038	0.0624	0.0576	0.0634	0.0732	0.0600	0.0656	0.0738	0.0835	0.1130	0.1450

Table 4.2: Classification error on Beijing Drivers with SVM.

kernel	statistics	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	$\mathtt{d}_{Q_1}$	$\mathtt{d}^{\pi}_{Q_1}$	Eu
linear	mean	0.2170	0.2066	0.2046	0.2173
	median	0.1987	0.1851	0.1892	0.2000
	SD	0.1185	0.1256	0.1221	0.1282
quadratic	mean	0.2327	0.2190	0.2000	0.2377
	median	0.2000	0.1778	0.1455	0.1949
	SD	0.1414	0.1678	0.1684	0.1668
Gaussian	mean	0.1725	0.0727	0.0733	0.0845
	median	0.1509	0.0587	0.0588	0.0690
	SD	0.1047	0.0594	0.0599	0.0670

Table 4.3: Classification error on Beijing with $|\tilde{Q}_1| = 200$.

	statistics	KNN	linear-SVM	quad-SVM	Gauss-SVM
$\mathtt{d}_{\tilde{Q}_1}$	mean	0.0801	0.1419	0.1398	0.0722
	median	0.0650	0.1125	0.0909	0.0581
	SD	0.0616	0.1058	0.1425	0.0591
$\mathtt{d}^{\pi}_{\tilde{Q}_1}$	mean	**0.0708**	0.1432	0.2606	0.0726
	median	0.0558	0.1179	0.2222	0.0583
	SD	0.0578	0.1022	0.1932	0.0597
$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_1}$	mean	0.1711	0.2362	0.2673	0.1735
	median	0.1471	0.2200	0.2460	0.1529
	SD	0.1041	0.1173	0.1455	0.1051

statistics is shown in Table 4.3, from which we can see for KNN, the performance of $\mathtt{d}_{\tilde{Q}_1}$ is better than Euclidean distance, and $\mathtt{d}^{\pi}_{\tilde{Q}_1}$ provides the smallest error (mean error 0.0708, smaller than 0.0714 of LCSS). Moreover, we can see as $|Q|$ increases, the error of $\mathtt{d}_{\tilde{Q}_1}$ and $\mathtt{d}^{\pi}_{\tilde{Q}_1}$ with three kernel functions all decrease, except $\mathtt{d}^{\pi}_{\tilde{Q}_1}$ with quadratic kernel. When we use quadratic kernel, the algorithm takes a long time to converge, and for $|Q| = 200$, the dimension of vectors used in $\mathtt{d}^{\pi}_{\tilde{Q}_1}$ is 400, so the algorithm may not converge within 200000 iterations. The relatively small improvement also demonstrates that even with a small size, random Q, the distances still perform at or near the state-of-the-art.

To show different Q sampled from the region can yield a similar result, we sample another

Figure 4.6: Left: the data set Q_2 (orange points), Right: the data set Q_3 (orange points).

four sets Q_2, Q_3 and $\tilde{Q}_2$, $\tilde{Q}_3$ according to uniform distribution, where Q_2, Q_3 are shown in Figure 4.6, and $|Q_2| = |Q_3| = 20$, $|\tilde{Q}_2| = |\tilde{Q}_3| = 200$.

The running result of different algorithms with different distances on Q_2, Q_3 and $\tilde{Q}_2$, $\tilde{Q}_3$ are shown in Table 4.4 and Table 4.5 respectively. From these two tables we can see different Q uniformly sampled from the region does not cause a large difference in statistics of classification errors, and for the case $|Q| = 200$ the result almost does not change.

4.4.4 Classifying Trajectories 2: Bus versus Car

As another example, we consider the GPS Trajectories Data Set [29] in UCI machine learning repository. There are 87 car trajectories, and 76 bus trajectories in Aracaju, a city of Brazil. We remove those trajectories having less than 10 critical points, and then 78 car trajectories and 45 bus trajectories are left. For these 123 trajectories are shown in Figure 4.7(Left), where pink curves are car trajectories and blue curves are bus trajectories. We hand-pick 10 points as Q_1 such that each point is close to one class of trajectories, and randomly generate 20 points as Q_2. Each time we randomly choose 23 car trajectories and 13 bus trajectories as test data, and use other trajectories as training data to perform classification experiments, and compute the error. We do this 1000 times and then compute the mean, median and standard deviation (SD) of the error for each algorithm.

The results are shown in Table 4.6, and we see the KNN classification results using all 14 distance, using either Q_1 (10 chosen near data) or Q_2 (20 randomly chosen). The results are slightly better for Q_2 in almost all distances $\mathtt{d}_Q$, $\mathtt{d}_Q^{\pi}$, $\mathrm{LSH1}_Q$, and $\mathrm{LSH2}_Q$ – except $\mathtt{d}_Q^{\leftrightarrow}$. In

Table 4.4: Classification error on Beijing Drivers with different Q ($|Q| = 20$)

	distance	mean	median	SD
KNN	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.1724	0.1493	0.1039
	$\mathtt{d}^{\leftrightarrow}_{Q_3}$	0.1670	0.1424	0.1029
	$\mathtt{d}_{Q_2}$	0.0816	0.0667	0.0620
	$\mathtt{d}_{Q_3}$	0.0802	0.0652	0.0625
	$\mathtt{d}^{\pi}_{Q_2}$	0.0721	0.0574	0.0573
	$\mathtt{d}^{\pi}_{Q_3}$	0.0697	0.0556	0.0566
	LSH1_{Q_2} (r=0.1)	0.1122	0.0860	0.0964
	LSH1_{Q_3} (r=0.1)	0.1190	0.0861	0.1110
	LSH2_{Q_2} (r=0.1)	0.2278	0.2027	0.1408
	LSH2_{Q_3} (r=0.07)	0.2070	0.1858	0.1256
linear SVM	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.2176	0.2000	0.1186
	$\mathtt{d}^{\leftrightarrow}_{Q_3}$	0.2145	0.1965	0.1173
	$\mathtt{d}_{Q_2}$	0.2039	0.1815	0.1250
	$\mathtt{d}_{Q_3}$	0.2041	0.1818	0.1246
	$\mathtt{d}^{\pi}_{Q_2}$	0.2005	0.1824	0.1213
	$\mathtt{d}^{\pi}_{Q_3}$	0.2018	0.1836	0.1215
quadratic SVM	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.2362	0.2034	0.1421
	$\mathtt{d}^{\leftrightarrow}_{Q_3}$	0.2277	0.1912	0.1438
	$\mathtt{d}_{Q_2}$	0.2155	0.1694	0.1698
	$\mathtt{d}_{Q_3}$	0.2152	0.1718	0.1678
	$\mathtt{d}^{\pi}_{Q_2}$	0.2009	0.1469	0.1694
	$\mathtt{d}^{\pi}_{Q_3}$	0.1932	0.1364	0.1682
Gaussian SVM	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.1751	0.1542	0.1057
	$\mathtt{d}^{\leftrightarrow}_{Q_3}$	0.1688	0.1470	0.1038
	$\mathtt{d}_{Q_2}$	0.0739	0.0595	0.0595
	$\mathtt{d}_{Q_3}$	0.0730	0.0583	0.0594
	$\mathtt{d}^{\pi}_{Q_2}$	0.0737	0.0595	0.0597
	$\mathtt{d}^{\pi}_{Q_3}$	0.0725	0.0580	0.0592

these experiments on Q_2, the best mean error (about 21% to 22%) is achieved by $\mathtt{d}^{\leftrightarrow}_{Q}$, $\mathtt{d}_{Q}$, and LSH1_{Q} (which required a parameter search). The best error is about 20% by $\mathtt{d}^{\leftrightarrow}_{Q}$ using Q_2. While $\mathtt{d}^{\pi}_{Q}$, LCSS, EDR, and LSH2_{Q} achieve error between 25% and 27%. Noticeably, the methods which were competitive with $\mathtt{d}_{Q}$ and $\mathtt{d}^{\pi}_{Q}$ on the Beijing Drivers data are EDR, which required a parameter tuned, as well as DTW and Eu, which now have error rate above 31%. As a baseline, always predicting "car" obtains 36% error.

We show the results of applying SVM in Table 4.6. Again the difference is small between Q_2 and Q_1. And while the linear and quadratic SVM do not perform that well; for the Gaussian kernel on $\mathtt{d}_{Q}$ and $\mathtt{d}^{\pi}_{Q}$ the mean error is only 16% to 20%, and 19% to 21% for $\mathtt{d}^{\leftrightarrow}_{Q}$. The overall best is $\mathtt{d}^{\pi}_{Q_1}$ achieving a mean error of 16.59%, a significant improvement over the KNN results.

Table 4.5: Classification error on Beijing Drivers with different Q ($|Q| = 200$)

	distance	mean	median	SD
KNN	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_2}$	0.1708	0.1469	0.1040
	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_3}$	0.1708	0.1471	0.1038
	$\mathtt{d}_{\tilde{Q}_2}$	0.0805	0.0652	0.0619
	$\mathtt{d}_{\tilde{Q}_3}$	0.0798	0.0645	0.0618
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.0707	0.0560	0.0571
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.0699	0.0556	0.0569
linear SVM	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_2}$	0.2353	0.2186	0.1173
	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_3}$	0.2357	0.2196	0.1173
	$\mathtt{d}_{\tilde{Q}_2}$	0.1425	0.1130	0.1061
	$\mathtt{d}_{\tilde{Q}_3}$	0.1414	0.1121	0.1055
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.1437	0.1189	0.1022
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.1429	0.1176	0.1019
quadratic SVM	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_2}$	0.2671	0.2447	0.1468
	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_3}$	0.2666	0.2455	0.1465
	$\mathtt{d}_{\tilde{Q}_2}$	0.1410	0.0909	0.1440
	$\mathtt{d}_{\tilde{Q}_3}$	0.1389	0.0895	0.1432
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.2615	0.2219	0.1937
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.2611	0.2215	0.1936
Gaussian SVM	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_2}$	0.1730	0.1521	0.1051
	$\mathtt{d}^{\leftrightarrow}_{\tilde{Q}_3}$	0.1734	0.1526	0.1052
	$\mathtt{d}_{\tilde{Q}_2}$	0.0725	0.0582	0.0592
	$\mathtt{d}_{\tilde{Q}_3}$	0.0719	0.0576	0.0589
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.0726	0.0583	0.0595
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.0721	0.0578	0.0592

4.4.5 Classifying Trajectories 3: Landmark-Sensitivity

To show the further advantage of $\mathtt{d}_Q$ and $\mathtt{d}^{\pi}_Q$, we create a synthetic data set that appears random, except one set of trajectories pass nearby a POI and the others do not. We randomly generate two classes of trajectories on the map of Beijing, and each class has 30 trajectories. Each trajectory has 10 critical points, and all blue trajectories passes through some point close to the city center, and all pink trajectories do not. We hand-pick a point at the Palace Museum, the center of the city, and randomly choose other 9 points to form the set Q. As shown in Figure 4.7(Right), these trajectories are a mess and largely indistinguishable, except that the blue set passes near the landmark: Palace Museum. We next show that $\mathtt{d}_Q$ and $\mathtt{d}^{\pi}_Q$ which are landmark-aware (e.g., POI-aware) have significantly more power in distinguishing these classes.

We randomly choose 21 trajectories from each class to form a training data set of size 42, and use the other trajectories as test data. Each time, we record the error, and repeat this 1000 times to output the mean, median and standard deviation (SD) of these errors.

Table 4.7 shows the KNN classification results. Distances Eu and dF provide no advantage over a random classifier (which would report error 0.5). $\mathtt{d}^{\pi}_{Q}$, $\mathtt{d}_{Q}$, $\mathtt{d}^{\leftrightarrow}_{Q}$, DTW, and Hausdorff

Table 4.6: Classification error on Bus vs. Car.

	distance	mean	median	SD
KNN	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	0.2027	0.1944	0.0647
	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.2148	0.2222	0.0624
	$\mathtt{d}_{Q_1}$	0.2331	0.2222	0.0669
	$\mathtt{d}_{Q_2}$	0.2229	0.2222	0.0637
	$\mathtt{d}^{\pi}_{Q_1}$	0.2608	0.2500	0.0625
	$\mathtt{d}^{\pi}_{Q_2}$	0.2505	0.2500	0.0627
	Eu	0.3323	0.3333	0.0661
	dF	0.3431	0.3333	0.0667
	DTW	0.3118	0.3056	0.0679
	dH	0.3284	0.3333	0.0627
	LCSS (ε=0.015,δ=3)	0.2448	0.2500	0.0605
	EDR (ε=0.015)	0.2640	0.2500	0.0622
	LSH1$_{Q_1}$ (r=0.02)	0.2673	0.2778	0.0448
	LSH2$_{Q_1}$ (r=0.08)	0.2516	0.2500	0.0467
	LSH1$_{Q_2}$ (r=0.03)	0.2209	0.2222	0.0622
	LSH2$_{Q_2}$ (r=0.05)	0.2690	0.2778	0.0464
linear SVM	Eu	0.3624	0.3611	0.0085
	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	0.3652	0.3611	0.0145
	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.3655	0.3611	0.0151
	$\mathtt{d}_{Q_1}$	0.3611	0.3611	0
	$\mathtt{d}_{Q_2}$	0.3611	0.3611	0
	$\mathtt{d}^{\pi}_{Q_1}$	0.3611	0.3611	0
	$\mathtt{d}^{\pi}_{Q_2}$	0.3612	0.3611	0.0018
quadratic SVM	Eu	0.3609	0.3611	0.0660
	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	0.3645	0.3611	0.0200
	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.3140	0.3056	0.0415
	$\mathtt{d}_{Q_1}$	0.3617	0.3611	0.0055
	$\mathtt{d}_{Q_2}$	0.3625	0.3611	0.0087
	$\mathtt{d}^{\pi}_{Q_1}$	0.2644	0.2500	0.0645
	$\mathtt{d}^{\pi}_{Q_2}$	0.2828	0.2778	0.0670
Gaussian SVM	Eu	0.2239	0.2222	0.0587
	$\mathtt{d}^{\leftrightarrow}_{Q_1}$	0.1940	0.1944	0.0554
	$\mathtt{d}^{\leftrightarrow}_{Q_2}$	0.2120	0.2222	0.0564
	$\mathtt{d}_{Q_1}$	0.1894	0.1944	0.0543
	$\mathtt{d}_{Q_2}$	0.1968	0.1944	0.0573
	$\mathtt{d}^{\pi}_{Q_1}$	**0.1659**	0.1667	0.0572
	$\mathtt{d}^{\pi}_{Q_2}$	0.1731	0.1667	0.0572

achieve only slight advantage over random classifiers, with error rates about 43% to 48%, with the best achieved by $\mathtt{d}_Q^{\pi}$. This extends to the SVM approaches in Table 4.8. The best

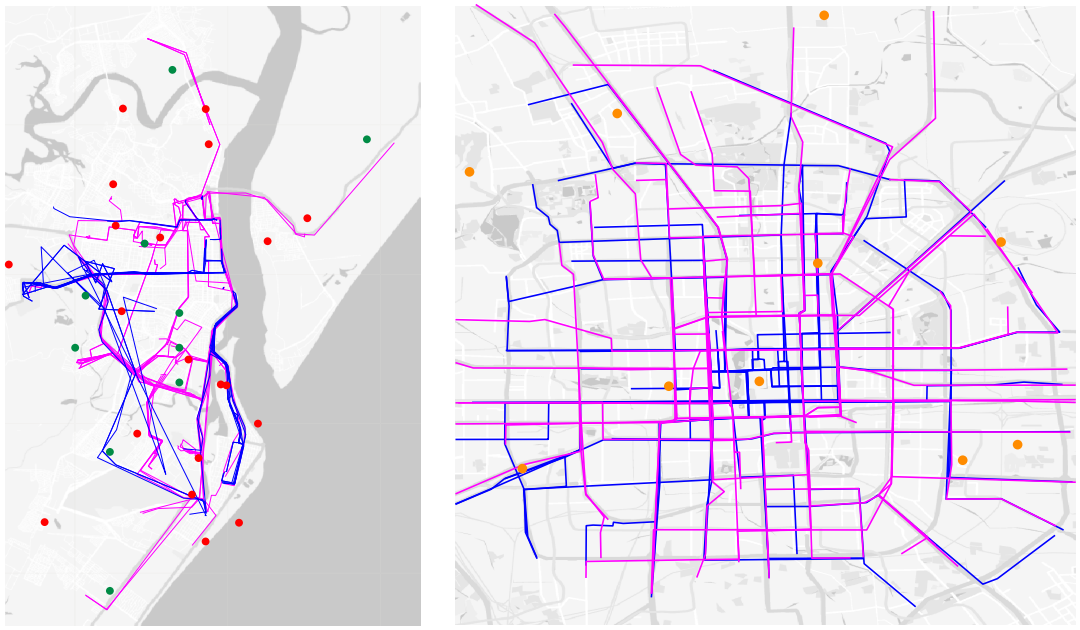

Figure 4.7: Left: Bus (blue) and car (pink) trajectories with landmark sets Q_1 (green points), Q_2 (red points). Right: Two classes of trajectories and Q (orange points).

Table 4.7: Landmark-sensitive classification error with KNN.

distance	mean	median	SD
Eu	0.5226	0.5000	0.0999
dF	0.5056	0.5000	0.0977
DTW	0.4777	0.5000	0.1033
dH	0.4627	0.4444	0.1025
LCSS ($\varepsilon = 0.001, \delta = 8$)	0.3437	0.3333	0.0812
EDR($\varepsilon = 0.02$)	0.3916	0.3889	0.0823
$\mathrm{LSH1}_Q$ (r=0.01)	0.2524	0.2222	0.0990
$\mathrm{LSH2}_Q$ (r=0.02)	0.3248	0.3333	0.0916
$\mathtt{d}_Q$	0.4729	0.5000	0.1005
$\mathtt{d}_{Q,W}$ ($w_1 = 0.3$)	0.4133	0.3889	0.1052
$\mathtt{d}_{Q,W}$ ($w_1 = 0.6$)	0.2687	0.2778	0.0969
$\mathtt{d}_{Q,W}$ ($w_1 = 0.9$)	0.0592	0.0556	0.0611
$\mathtt{d}_Q^{\pi}$	0.4385	0.4444	0.0961
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.3$)	0.3846	0.3889	0.0921
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.6$)	0.2396	0.2222	0.0804
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.9$)	0.1002	0.0556	0.0817
$\mathtt{d}_Q^{\leftrightarrow}$	0.4711	0.4444	0.1027
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.3$)	0.4468	0.4444	0.1062
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.6$)	0.4377	0.4444	0.1060
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.9$)	0.4466	0.4444	0.1002

parameter free approach is $\mathtt{d}_Q^{\pi}$ at 43.85% error. The parameterized distances LCSS, EDR, LSH1$_Q$, and LSH2$_Q$ perform better with error rates 25% to 40%; but these can be sensitive to the parameter choices – we only show the best results.

Next we can consider re-weighting the importance of the landmarks Q, for instance in the case where one particular POI (in this case q_1) is known to have a specific meaning in the classification task (e.g., did someone stop by the sporting event, or a military point of interest). Suppose $w_i > 0$ is a weight of $q_i \in Q$, and $W = (w_1, w_2, ..., w_n)$. Then we can generalize the definitions to:

$$\mathtt{d}_{Q,W}(\gamma^{(1)}, \gamma^{(2)}) = \left(\sum_{i=1}^{n} w_i \big(d_i^{(1)} - d_i^{(2)}\big)^2 \right)^{\frac{1}{2}}, \;\; \mathtt{d}_{Q,W}^{\pi}(\gamma^{(1)}, \gamma^{(2)}) = \sum_{i=1}^{n} w_i \big(\|p_i^{(1)} - p_i^{(2)}\|\big).$$

Let $w_1 \in (0, 1)$ be the weight of q_1, and $w_i = \frac{1}{9}(1 - w_1)$ (for $2 \leq i \leq 10$) be the weight of all other points in Q.

Now observe in Table 4.7 that the landmark-based distance using a KNN classifier can achieve very low error (6% for $\mathtt{d}_{Q,W}$ and 10% for $\mathtt{d}_{Q,W}^{\pi}$) as we gradually increase the weight

Table 4.8: Landmark-sensitive classification error with SVM.

kernel	statistics	$\mathtt{d}_Q^{\leftrightarrow}$	$\mathtt{d}_Q$	$\mathtt{d}_Q^{\pi}$	Eu
linear	mean	0.5000	0.4586	0.4941	0.5887
	median	0.5000	0.4444	0.5000	0.6111
	SD	0.0976	0.0983	0.0997	0.0925
quadratic	mean	0.5403	0.4617	0.5574	0.4795
	median	0.5556	0.4444	0.5556	0.5000
	SD	0.0957	0.0967	0.1007	0.1059
Gaussian	mean	0.5059	0.4567	0.4556	0.5906
	median	0.5000	0.4444	0.4444	0.6111
	SD	0.0959	0.0944	0.0997	0.0939

Table 4.9: Landmark-sensitive classification error with weighted Gaussian SVM.

metrics	mean	median	SD
$\mathtt{d}_{Q,W}$ ($w_1 = 0.3$)	0.1487	0.1667	0.0809
$\mathtt{d}_{Q,W}$ ($w_1 = 0.6$)	0.0303	0	0.0369
$\mathtt{d}_{Q,W}$ ($w_1 = 0.9$)	**0.0159**	0	0.0256
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.3$)	0.2997	0.2778	0.0937
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.6$)	0.1053	0.1111	0.0702
$\mathtt{d}_{Q,W}^{\pi}$ ($w_1 = 0.9$)	0.0316	0	0.0386
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.3$)	0.4942	0.5000	0.0977
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.6$)	0.4726	0.5000	0.0976
$\mathtt{d}_{Q,W}^{\leftrightarrow}$ ($w_1 = 0.9$)	0.4687	0.4444	0.0974

Table 4.10: Landmark-sensitive classification error with weighted linear SVM.

metrics	mean	median	SD
$\mathtt{d}_{Q,W}$ ($w_1 = 0.3$)	0.3309	0.3333	0.0836
$\mathtt{d}_{Q,W}$ ($w_1 = 0.6$)	0.3083	0.3333	0.1019
$\mathtt{d}_{Q,W}$ ($w_1 = 0.9$)	0.3051	0.3333	0.1089
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.3$)	0.4936	0.5000	0.0903
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.6$)	0.4191	0.4444	0.0700
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.9$)	0.4104	0.3889	0.0694
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.3$)	0.4372	0.4444	0.0901
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.6$)	0.4340	0.4444	0.0893
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.9$)	0.4329	0.4444	0.0895

Table 4.11: Landmark-sensitive classification error with weighted quadratic SVM.

metrics	mean	median	SD
$\mathtt{d}_{Q,W}$ ($w_1 = 0.3$)	0.3309	0.3333	0.0836
$\mathtt{d}_{Q,W}$ ($w_1 = 0.6$)	0.3084	0.3333	0.1019
$\mathtt{d}_{Q,W}$ ($w_1 = 0.9$)	0.3051	0.3333	0.1089
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.3$)	0.5302	0.5000	0.0992
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.6$)	0.5270	0.5000	0.1023
$\mathtt{d}^{\pi}_{Q,W}$ ($w_1 = 0.9$)	0.3909	0.3889	0.0773
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.3$)	0.4367	0.4444	0.0902
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.6$)	0.4333	0.4444	0.0892
$\mathtt{d}^{\leftrightarrow}_{Q,W}$ ($w_1 = 0.9$)	0.4322	0.4444	0.0889

of the point q_1 from $w_1 = 0.1$ (i.e., $\mathtt{d}_Q$ or $\mathtt{d}^{\pi}_Q$) to $w_1 = 0.9$ to emphasize a desired POI. The result is even more pronounced for the Gaussian SVM, as shown in Table 4.9; similar plots are shown for linear and quadratic kernels in Table 4.10 and Table 4.11. As w_1 is increased from (uniform) 0.1 to 0.9, the mean error decreases from 45% to 1.5% for $\mathtt{d}_{Q,W}$ and from 45% to 3% for $\mathtt{d}^{\pi}_{Q,W}$. Thus, while all other distances we tried are only slightly better than random unless their parameters are tuned, by emphasizing a particular POI (a very intuitive adjustment), we achieve almost no error in classifying these trajectories.

4.4.6 Using $\mathtt{d}_Q$ in Nearest Neighbor Search

We demonstrate that $\mathtt{d}_Q$'s sketched representation of the trajectories in $\mathbb{R}^{|Q|}$ allows for *extremely efficient k-nearest neighbor search*. We consider two representative methods [80, 92] for comparison; but do all, e.g., [39]) which require timing information.

As a first comparison, consider a recent heavily-optimized kNN search algorithm focusing on Hausdorff and dF distances [92]; this system, DFT, is optimized for distributed algorithms

on a cluster, but show results on 1 node which we compare against. We obtained a random sample of the GEN-TRAJ data set containing $m = 3$ million trajectories, using 36GB of storage (larger than their 30.9GB dataset [92]). From *their* Figure 10, their indexes take 2000 to 6000 sec to build, and kNN queries require 50 to 200 seconds for $k = 10$.

Another distributed system DITA [80] for trajectory similarity search focuses on DTW, returning all trajectories within a threshold. In their [80] Figures 7(a) and 8(a), using 256 cores they achieve query time between 0.001 and 0.01 seconds on Beijing (10.4GB) and Chengdu (28GB) datasets.

To perform kNN queries using $\mathtt{d}_Q$ we can sketch trajectories as $|Q|$-dimensional vectors and use Euclidean distance. Hence, once we create the sketches, we can use any of the highly optimized packages for kNN Euclidean queries (c.f., `http://ann-benchmarks.com`); we choose a consistent top performer K-Graph (`https://github.com/aaalgo/kgraph`) with settings: recall=0.99 and max_iteration=50. We run on a desktop with a 6-core Intel Xeon CPU ES-1650 v3 @3.5GHz processor, and 128GB RAM; the same processor as in DFT [92].

For experiments, we randomly choose a set of landmarks among the trajectories with $|Q| = \{12, 20, 28, 36, 44, 52\}$. From these Q we preprocess the data to derive $m \times |Q|$ sketches, a txt file we pass to K-Graph. Then K-Graph builds an index, and allows queries. The preprocessing time (to build sketch), sketch file size, time to build K-Graph's index, that index size, and the average query time are shown in Table 4.12. For all these different values of $|Q|$, the K-Graph algorithm reaches recall=0.99 within 7 iterations.

Table 4.12: The running time experiment of KNN search.

$\|Q\|$	12	20	28	36	44	52
preprocessing time (s)	38	62	88	114	138	160
sketch size (MB)	337	560	785	1012	1331	1536
index time (s)	106	109	114	119	124	129
index file size (MB)	999	999	1005	1002	1007	1001
query time (10^{-4}s)	4.2	3.7	4.2	3.2	3.5	3.7

The preprocessing and index building times take 38 to 160 seconds and 106 to 129 seconds, respectively. By comparison, it takes 673 seconds to load the raw data into memory. Combined they are an order of magnitude faster than the index build time for Hausdorff in DFT [92]. The sketch size is only 300 to 1500 MB, and the index sizes are 1000 MB;

reducing the size by 1 or 2 orders of magnitude from the original size. Finally, the query times are only 0.00032 to 0.00042 seconds; that is 5 orders of magnitude faster than the DFT index optimized for Hausdorff distance! and 1 to 2 orders of magnitude faster than DITA optimized for DTW and using 256 cores on smaller data. Thus, using $\mathtt{d}_Q$ (and existing libraries) allows for small data sketches, and extremely efficient kNN queries.

4.4.7 Online Data and Code

The experiments in Section 4.4.2, Section 4.4.3, Section 4.4.4 and Section 4.4.5 are similar to the experiments in [76], where a link of data and code is given. The raw data, intermediate data and code to reproduce the result of experiments in Appendix B.4 are available here:
`https://drive.google.com/open?id=1OAu5yoSH6MMlaBCkhgll43THxRpUb7Qm`

4.5 Discussion

On trajectories, new metrics $\mathtt{d}_Q$ and $\mathtt{d}_Q^\pi$ are the most general and best or competitive against all other distances in *all* analysis tasks; see Table 4.13. LCSS performs better under some other conditions for Beijing drivers experiment, see Appendix B.4. However from Table 4.13 we can see $\mathtt{d}_Q$ and $\mathtt{d}_Q^\pi$ are either the best or nearly best at each task, especially when the computation cost is considered for each distance. The main point that $\mathtt{d}_Q$ and $\mathtt{d}_Q^\pi$ are the consistently among the best should hold under any reasonable subjective way to alter how this summary is presented.

Table 4.13: Distances on analysis tasks as: best ●, competitive ⬤, near competitive ○; possible ✓ or possible but slower ✓.

task	$\mathtt{d}_Q$	$\mathtt{d}_Q^\pi$	$\mathtt{d}_Q^\leftrightarrow$	Eu	dF	DTW	dH	LCSS	EDR	LSH_Q
easy clustering	✓	✓	✓	✓	-	-	-	-	-	-
learn 1	⬤	●	-	⬤	○	⬤	⬤	⬤	⬤	-
learn 2	⬤	●	○	○	-	⬤	-	○	⬤	○
learn 3	●	⬤	-	-	-	-	-	-	-	-
fast NN	✓	✓	✓	✓	-	✓	-	-	-	✓
any k	✓	✓	-	-	✓	✓	✓	✓	✓	✓

The landmark set Q can be randomly chosen and small, or its points can hold specific meaning in which case, the interpretation and discriminatory ability of the distances are greatly enhanced. Chapter 5 provides an in depth theoretical study of how many landmarks

are required to preserve certain errors, how to chose them, and when curves can be explicitly recovered from them. In this work, we simply empirically show that in most cases 20 random landmarks are sufficient.

These provide meaningful *vectorized representations*. They are general and simple to compute and work with. We believe many applications of these sorts of vectorized distances will be discovered. And there are more mathematical questions to ask about the geometric and statistical power of these landmark-based distances.

CHAPTER 5

SKETCHED MINDIST

5.1 Introduction

In this chapter we generalize $\mathtt{d}_Q$ in Chapter 4 to general geometric objects. For an object $J \in \mathcal{J}$, where $J \subset \mathbb{R}^d$, this depends on a set of *landmarks* $Q \subset \mathbb{R}^d$; for now let $n = |Q|$. These landmarks induce a *sketched representation* $v_Q(J) \in \mathbb{R}^n$ where the ith coordinate $v_i(J)$ is defined via a MinDist operation

$$v_i(J) = \mathrm{dist}(q_i, J) = \inf_{p \in J} \|p - q_i\|,$$

using the ith landmark $q_i \in Q$. When the object J is implicit, we simply use v_i. Then our new distance $\mathtt{d}_Q$ between two objects $J_1, J_2 \in \mathcal{J}$ is simply the (normalized) Euclidean distance between the sketched representations

$$\mathtt{d}_Q(J_1, J_2) = \left\|\bar{v}_Q(J_1) - \bar{v}_Q(J_2)\right\|,$$

where $\bar{v}_Q = \frac{1}{\sqrt{|Q|}} v_Q$.

Chapter 4 introduces other variants of this distance (using other norms or using the $\arg\min_{p \in J}$ points on each $J \in \mathcal{J}$). We focus on this version as it is the simplest, cleanest, easiest to use, and was the best or competitive with the best on all empirical tasks. Indeed, for the pressing case of measuring a distance between trajectories, this new distance measure dominates a dozen other distance measures (including dynamic time warping, discrete Frechet distance, edit distance for real sequences) in terms of classification performance, and is considerably more efficient in clustering and nearest neighbor tasks.

The goal of this chapter is to formally understand how many landmarks in Q are needed for various error guarantees, and how to chose the locations of these points Q.

Our aims in the choice of Q are two-fold: first, we would like to approximate $\mathtt{d}_Q$ with $\mathtt{d}_{\tilde{Q}}$, and second we would like to recover $J \in \mathcal{J}$ exactly only using $v_Q(J)$. The specific results

vary depending on the initial set Q and the object class $\mathcal{J}$. More precisely, the approximation goal aims to preserve d_Q for all objects J in some class $\mathcal{J}$ with a subset $\tilde{Q} \subset Q$ of landmarks. Or possibly a weighted set of landmarks $W, \tilde{Q}$ with $|\tilde{Q}| = N$, so each q_i is associated with a weight w_i and the weighted distance is defined

$$\mathtt{d}_{\tilde{Q},W}(J_1, J_2) = \sqrt{\sum_{i=1}^{N} w_i \cdot (v_i(J_1) - v_i(J_2))^2} = \left\| \tilde{v}_{\tilde{Q}}(J_1) - \tilde{v}_{\tilde{Q}}(J_2) \right\|.$$

where $\tilde{v}_{\tilde{Q}} = (\tilde{v}_1, \cdots, \tilde{v}_N)$ with $\tilde{v}_i = \sqrt{w_i} v_i$. Specifically, our aim is an $(\rho, \varepsilon, \delta)$-*approximation of* Q *over* $\mathcal{J}$ so when $W, \tilde{Q}$ is selected by a random process that succeeds with probability at least $1 - \delta$, then for a *pair* $J_1, J_2 \in \mathcal{J}$ with $\mathtt{d}_Q(J_1, J_2) \geq \rho$

$$(1 - \varepsilon)\mathtt{d}_Q(J_1, J_2) \leq \mathtt{d}_{\tilde{Q},W}(J_1, J_2) \leq (1 + \varepsilon)\mathtt{d}_Q(J_1, J_2).$$

When this holds for *all* pairs in $\mathcal{J}$, we say it is a *strong* $(\rho, \varepsilon, \delta)$-*approximation of* Q *over* $\mathcal{J}$. In some cases we can set to 0 either δ (the process is deterministic) or ρ (this preserves even arbitrarily small distances), and may be able to use uniform weights $w_i = \frac{1}{|\tilde{Q}|}$ for all selected points.

5.1.1 Our Results

We begin with a special signed variant of the distance associated with the class $\mathcal{J}$ of $(d-1)$-dimensional hyperplanes (which for instance could model linear separators or linear regression models). The signed variant provides $v_i(J)$ a negative value on one side of the separator. In this variant, we show that if Q is full rank, then we can recover J from $v_Q(J)$, and a variant of sensitivity sampling can be used to select $O(d/(\delta\varepsilon^2))$ points to provide a $(0, \varepsilon, \delta)$-approximation $W, \tilde{Q}$. Or by selecting $O(\frac{d}{\varepsilon^2}(d \log d + \log \frac{1}{\delta}))$ results in a strong $O(0, \varepsilon, \delta)$-approximation (Theorem 5.4).

Next we consider the more general case where the objects are bounded geometric objects $\mathcal{S}$. For such objects it is useful to consider a bounded domain $\Omega_L = [0, L]^d$ (for d a fixed constant), and consider the case where each $S \in \mathcal{S}$ and landmarks satisfy $S, Q \subset \Omega_L$. In this case, the number of samples required for a $(\rho, \varepsilon, \delta)$-approximation is $\mathfrak{S}_Q \frac{1}{\varepsilon^2 \delta}$ where

$$\mathfrak{S}_Q = O\left(\left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}} \min\left(\log \frac{L}{\eta}, \log n, \left(\frac{L}{\rho}\right)^2 \right)^{\frac{2}{2+d}} \right), \tag{5.1}$$

where $\eta = \min_{q,q' \in Q} \|q - q'\|_\infty$. A few special cases are worth expanding upon. When Q is continuous and uniform over Ω_L then $\mathfrak{S}_Q = O((L/\rho)^{\frac{2d}{2+d}})$, and this is tight in $\mathbb{R}^2$ at

$\mathfrak{S}_Q = \Theta(L/\rho)$. That is, we can show that $\mathfrak{S}_Q = \Theta(L/\rho)$ may be needed in general. When $d = 2$ but not necessarily uniform on Ω_L, then $\mathfrak{S}_Q = O(\frac{L}{\rho}\min\{\sqrt{\log n}, L/\rho\})$. And when Q is on a grid over Ω_L in $\mathbb{R}^2$ of resolution $\Theta(\rho)$, then $\mathfrak{S}_Q = O(\frac{L}{\rho}\sqrt{\log \frac{L}{\rho}})$, just a $\sqrt{\log L/\rho}$ factor more than the lower bound.

We conclude with some specific results for trajectories. When considering the class $\mathcal{T}_k$ with at most k segments, then $O(\frac{1}{\varepsilon^2}\mathfrak{S}_Q(k^3 \log \mathfrak{S}_Q + \log \frac{1}{\delta}))$ samples is sufficient for a *strong* $(\rho, \varepsilon, \delta)$-approximation. Then when considering trajectories $\mathcal{T}_\tau$ where the critical points are at distance at least τ apart from any non-adjacent part of the curve, we can *exactly reconstruct* the trajectory from v_Q as long as Q is a grid of side length $\Omega(\tau)$. It is much cleaner to describe the results for trajectories and Q precisely on a grid, but these results should extend for any object with k piecewise-linear boundaries, and critical points sufficiently separated, or Q as having any point in each sufficiently dense grid cell, as opposed to being exactly on the grid lattice.

5.1.2 Connections to other Domains, and Core Challenges

Before deriving these results, it is useful to lay out the connection to related techniques, including ones that our results will build on, and the challenges in applying them.

Sensitivity sampling. Sensitivity sampling [40, 44, 63, 86] is an important technique for our results. This typically considers a dataset X (a subset of a metric space), endowed with a measure $\mu : X \to \mathbb{R}^+$, and a family of cost functions F. These cost functions are usually related to the fitting of a data model or a shape S to X, and for instance on a single point $x \in X$, for $f \in F$, where

$$f(x) = \mathrm{dist}(x, S)^2 = \inf_{p \in S} \|x - p\|^2$$

is the squared distance from x to the closest point p on the shape S. And then $\bar{f} = \int_X f(x)\mathrm{d}\mu(x)$. The *sensitivity* [63] of $x \in X$ w.r.t. (F, X, μ) is defined as:

$$\sigma_{F,X,\mu}(x) := \sup_{f \in F} \frac{f(x)}{\bar{f}},$$

and the *total sensitivity* of F is defined as: $\mathfrak{S}(F) = \int_X \sigma_{F,X,\mu}(x)\mathrm{d}\mu(x)$. This concept is quite general, and has been widely used in applications ranging from various forms of clustering [40, 44] to dimensionality reduction [43] to shape-fitting [86]. In particular, this will allow us to draw N samples $\tilde{X}$ iid from X proportional to $\sigma_{F,X,\mu}(x)$, and weighted

$\tilde{w}(\tilde{x}) = \frac{\mathfrak{S}(F)}{N \cdot \sigma_{F,X,\mu}(\tilde{x})}$; we call this $\sigma_{F,X,\mu}$-*sensitive sampling*. Then $\tilde{X}$ is a $(0, \varepsilon, \delta)$-coreset; that is, with probability $1 - \delta$ for *each* $f \in F$

$$(1-\varepsilon)\bar{f} \leq \int_{\tilde{X}} f(\tilde{x}) \mathrm{d}\tilde{w}(\tilde{x}) \leq (1+\varepsilon)\bar{f},$$

using $N = O(\frac{\mathfrak{S}(F)}{\varepsilon^2 \delta})$ [63]. The same error bound holds for *all* $f \in F$ (then it is called a $(0, \varepsilon, \delta)$-*strong coreset*) with $N = O(\frac{\mathfrak{S}(F)}{\varepsilon^2}(\mathfrak{s}_F \log \mathfrak{S}(F) + \log \frac{1}{\delta}))$ where $\mathfrak{s}_F$ is the shattering dimension of the range space $(X, \mathsf{ranges}(F))$ [16]. Specifically, each range $r \in (X, \mathsf{ranges}(F))$ is defined as those points in a sublevel set of a specific cost function $r = \{x \in X \mid \frac{\mu(x)}{\mathfrak{S}(F)} \frac{f(x)}{\bar{f}} \leq \xi\}$ for some $f \in F$ and $\xi \in \mathbb{R}$.

It seems natural that a form of our results would follow directly from these approaches. However, two significant and intertwined challenges remain. First, our goal is to approximate the distance between a pair of sketches $\|v_Q(J_1) - v_Q(J_2)\|$, where these results effectively only preserve the norm of a single sketch $\|v_Q(J_1)\|$; this prohibits many of the geometric arguments in the prior work on this subject. Second, the total sensitivity $\mathfrak{S}(F)$ associated with unrestricted Q and pairs $J_1, J_2 \in \mathcal{J}$ is in general unbounded (as we prove in Lemma 5.4). Indeed, if the total sensitivity was bounded, it would imply a mapping to bounded vector space [63], wherein the subtraction of the two sketches $v_Q(J_1) - v_Q(J_2)$ would still be an element of this space, and the norm bound would be sufficient.

We circumvent these challenges in two ways. First, we identify a special case in Section 5.2 (with negative distances, for hyperplanes) under which there is a mapping of the sketch $v_Q(J_1)$ to metric space independent of the size and structure of Q. This induces a bound for total sensitivity related to a single object, and allows the subtraction of two sketches to be handled within the same framework.

Second, we enforce a lower bound on the distance $\mathtt{d}_Q(J_1, J_2) > \rho$ and an upper bound on the domain $\Omega_L = [0, L]^d$. This induces a restricted class of pairs $\mathcal{J}_{L/\rho}$ where L/ρ is a scaleless parameter, and it shows up in bounds we are then able to produce for the total sensitivity with respect to $\mathcal{J}_{L/\rho}$ and $Q \subset \Omega_L$.

Leverage scores, and large scales. Let $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse of a matrix, so $(AA^T)^+ = (AA^T)^{-1}$ when AA^T is full rank. The *leverage score* [36] of the ith column a_i of matrix A is defined as: $\tau_i(A) := a_i^T (AA^T)^+ a_i$. This definition is more specific

and linear-algebraic than sensitivity, but has received more attention for scalable algorithm development and approximation [15, 25, 26, 35, 36, 74].

However, Theorem 5.2 (in the Appendix 5.2.3) shows that if F is the collection of some functions defined on a set Q of n points ($\mu(q_i) = \frac{1}{n}$ for all $q_i \in Q$), where each $f \in F$ is the square of some function v in a finite dimensional space V spanned by a basis $\{v^{(1)}, \cdots, v^{(\kappa)}\}$, then we can build a $\kappa \times n$ matrix A where the ith column is $\frac{1}{\sqrt{n}}\left(v^{(1)}(q_i), \cdots, v^{(\kappa)}(q_i)\right)^T$, and have $\frac{1}{n} \cdot \sigma_{F,Q,\mu}(q_i)$ is precisely the leverage score of the ith column of the matrix A. A similar observation has been made by Varadarajan and Xiao [86].

A concrete implication of this connection is that we can invoke an online row sampling algorithm of Cohen *et.al.* [26]. In our context, this algorithm would stream over Q, maintaining (ridge) estimates of the sensitivity of each q_i from a sample $\tilde{Q}_{i-1}$, and retaining each q_i in that sample based on this estimate. Even in this streaming setting, this provides an approximation bound not much weaker than the sampling or gridding bounds we present; see Appendix 5.2.3.

Connection from MinDist to shape reconstruction. The fields of computational topology and surface modeling have extensively explored [18, 19, 77] the distance function to a compact set $J \subset \mathbb{R}^d$

$$\mathtt{d}_J(x) = \mathrm{dist}(x, J) = \inf_{p \in J} \|x - p\|,$$

their approximations, and the offsets $J^r = \mathtt{d}_J^{-1}([0, r])$. For instance the Hausdorff distance between two compact sets J, J' is $\mathtt{d}_{\mathsf{H}}(J, J') = \|\mathtt{d}_J - \mathtt{d}_{J'}\|_\infty$. The gradient of $\mathtt{d}_J$ implies stability properties about the medial axis [20]. And most notably, this stability of $\mathtt{d}_J$ with respect to a sample $P \sim J$ or $P \sim \partial J$ is closely tied to the development of shape reconstruction (aka geometric and topological inference) through α-shapes [37], power crust [10], and the like. The intuitive formulation of this problem through $\mathtt{d}_J$ (as opposed to Voronoi diagrams of P) has led to more statistically robust variants [19, 77] which also provide guarantees in shape recovery up to small feature size [45], essentially depending on the maximum curvature of ∂J.

Our formulation flips this around. Instead of considering samples P from J (or ∂J) we consider samples Q from some domain $\Omega \subset \mathbb{R}^d$. This leads to new but similar sampling theory, still depending on some feature size (represented by various scale parameters ρ, τ,

and η), and still allowing recovery properties of the underlying objects. While the samples P from J can be used to estimate Hausdorff distance via an all-pairs $O(|P|^2)$-time comparison, our formulation requires only a $O(|Q|)$-time comparison to compute $\mathtt{d}_Q$. We leave as open questions the recovering of topological information about an object $J \in \mathcal{J}$ from $v_Q(J)$.

Function space sketching. While most geometric inference sampling bounds focus on low-level geometric parameters (e.g., weak local feature size, etc), a variant based on the kernel distance $\mathtt{d}_K(P, x)$ [77] can be approximated (including useful level sets) using a uniform sample $P' \sim P$. The kernel distance in this setting is defined $\mathtt{d}_K(P, x) = \sqrt{1 + \mu_K(P) - 2\mathrm{KDE}_P(x)}$ where the kernel density estimate is defined $\mathrm{KDE}_P(x) = \frac{1}{|P|}\sum_{p \in P} K(p, x)$ with $K(p, x) = \exp(-\|x - p\|^2)$ and $\mu_K(P) = \frac{1}{P}\sum_{p \in P} \mathrm{KDE}_P(p)$. This sampling mechanism can be used to analyze KDE_P (and thus also $\mathtt{d}_K$) [73] by considering a reproducing kernel Hilbert space (RKHS) $\mathcal{H}_K$ associated with K; this is a function space so each element $\phi_K(p) = K(p, \cdot) \in \mathcal{H}_K$ is a function. And averages $\Phi_K(P) = \frac{1}{P}\sum_{p \in P} \phi_K(p) = \mathrm{KDE}_P$ are kernel density estimates. Ultimately, $O(\frac{1}{\varepsilon^2}\log\frac{1}{\delta})$ samples $\tilde{P}$ yields [68] with probability $1 - \delta$ that $\|\Phi_K(P) - \Phi_K(\tilde{P})\|_{\mathcal{H}_K} \leq \varepsilon$ which implies $\|\mathrm{KDE}_P - \mathrm{KDE}_{\tilde{P}}\|_\infty \leq \varepsilon$, and hence also $\|\mathtt{d}_K(P, \cdot) - \mathtt{d}_K(\tilde{P}, \cdot)\|_\infty \leq \Theta(\sqrt{\varepsilon})$. Notably, the natural $\mathcal{H}_K$-norm is an ℓ_2-norm when restricted to any finite dimensional subspace (e.g., the basis defined by $\{\phi_K(p)\}_{p \in P}$).

Similarly, our approximations of $\mathtt{d}_Q(\cdot, \cdot)$ using a sample $\tilde{Q} \sim Q$ result in a similar function space approximation. Again the main difference is that $\mathtt{d}_Q$ is bivariate (so it takes in a pair $J_1, J_2 \in \mathcal{J}$, which is hard to interpret geometrically), and we seek a relative error (not an additive error). This connection leads us to realize that there are JL-type approximations [59] of this feature space. That is, given a set of t objects $O = J_1, J_2, \ldots, J_t \subset \mathcal{J}$, and their representations $v_Q(J_1), v_Q(J_2), \ldots, v_Q(J_t) \in \mathbb{R}^n$, there is a mapping h to $\mathbb{R}^N$ with $N = O((1/\varepsilon^2)\log\frac{t}{\delta})$, so with probability at least $1 - \delta$ so for any pair $J, J' \in O$ $(1 - \varepsilon)\mathtt{d}_Q(J, J') \leq \|h(v_Q(J) - h(v_Q(J')\| \leq (1 + \varepsilon)\mathtt{d}_Q(J, J')$. However, for such a result to hold for *all* pairs in $\mathcal{J}$, there likely requires a lower bound on the distance ρ and/or upper bound on the underlying space L, as with the kernels [22, 75]. Moreover, such an approach would not provide an explicit coreset $\tilde{Q}$ that is interpretably in the original space $\mathbb{R}^d$.

5.2 The Distance Between Two Hyperplanes

In this section, we assume for two hyperplanes $h_1, h_2 \in \mathcal{H}$, $\mathtt{d}_Q$ is defined by (4.2) and study how to efficiently compute $\mathtt{d}_Q$ approximately, when the data set Q is very large. The basic idea is to use the sensitivity sampling method [63], and an online row sampling algorithm designed for leverage sampling [36].

5.2.1 Estimation of $\mathtt{d}_Q$ by Sensitivity Sampling on Q

Suppose $Q = \{q_1, q_2, \cdots, q_n\} \subset \mathbb{R}^d$, where q_i has the coordinate $(x_{i,1}, x_{i,2}. \cdots, x_{i,d})$. Without specification, in this chapter Q is a multiset, which means two points in Q can be at the same location, and $\|\cdot\|$ represents l^2 norm.

Any hyperplane $h \in \mathcal{H}$ can be uniquely expressed in the form

$$h = \left\{x = (x_1, \cdots, x_d) \in \mathbb{R}^d \mid \sum_{j=1}^{d} u_j x_j + u_{d+1} = 0\right\},$$

where $(u_1, \cdots, u_{d+1})$ is a vector in $\mathbb{U}^{d+1}$ defined in Section 4.2.2. A sketched halfspace h has n-dimensional vector $v_Q(h) = (v_1(h), \ldots, v_n(h))$ where each coordinate v_i is defined as the *signed* distance from q_i to the closest points on h, which can be calculated $v_i(h) = \sum_{j=1}^{d} u_j x_{i,j} + u_{d+1}$; the dot-product with the unit normal of h, plus offset u_{d+1}. As before, the distance is defined as $d_Q(h_1, h_2) = \|\frac{1}{\sqrt{n}}(v_Q(h_1) - v_Q(h_2))\|$.

When $Q \subset \mathbb{R}^d$ is *full rank* – that is, there are $d+1$ points in Q which are not on a common hyperplane – then Chapter 4 shows d_Q is a metric on $\mathcal{H}$.

We use sensitivity sampling to estimate $\mathtt{d}_Q$ with respect to a tuple (F, X, μ). First suppose $Q = \{q_1, \cdots, q_n\} \subset \mathbb{R}^d$ is full rank and $n \geq d+1$. Then we can let $X = Q$ and $\mu = \frac{1}{n}$; what remains is to define the appropriate F. Roughly, F is defined with respect to a $(d+1)$-dimensional vector space V, where for each $f \in F$, $f = v^2$ for some $v \in V$; and V is the set of all linear functions on $x \in Q$.

We now define F in more detail. Recall each $h \in \mathcal{H}$ can be represented as a vector $u \in \mathbb{U}^{d+1}$. This u defines a function $v_u(q) = \sum_{i=1}^{d} u_i x_i + u_{d+1}$, and these functions are elements of V. The vector space is however larger and defined

$$\begin{aligned} V = \{v_a : Q \mapsto \mathbb{R} \mid v_a(q) = \sum_{i=1}^{d} a_i x_i + a_{d+1} \text{ where } q = (x_1, \cdots, x_d) \in Q, \\ a = (a_1 \cdots, a_{d+1}) \in \mathbb{R}^{d+1}\}, \end{aligned}$$

so that there can be $v_a \in V$ for which $a \notin \mathbb{U}^{d+1}$; rather it can more generally be in $\mathbb{R}^{d+1}$. Then the desired family of real-valued functions is defined

$$F = \{f : Q \mapsto [0, \infty) \mid \exists\, v \in V \text{ s.t. } f(q) = v(q)^2,\ \forall q \in Q\}.$$

To see how this can be applied to estimate $\mathtt{d}_Q$, consider two hyperplanes h_1, h_2 in $\mathbb{R}^d$ and the two unique vectors $u^{(1)}, u^{(2)} \in \mathbb{U}^{d+1}$ which represent them. Now introduce the vector $u = (u_1, \cdots, u_{d+1}) = u^{(1)} - u^{(2)}$; note that $u \in \mathbb{R}^{d+1}$, but not necessarily in $\mathbb{U}^{d+1}$. Now for $q \in Q$ define a function $f_{h_1,h_2} \in F$ as

$$f_{h_1,h_2}(q) = f_{h_1,h_2}(x_1, \cdots, x_d) = \big(\textstyle\sum_{i=1}^d u_i x_i + u_{d+1}\big)^2,$$

so $\mathtt{d}_Q(h_1, h_2) = (\frac{1}{n}\sum_{q\in Q} f_{h1,h2}(q))^{\frac{1}{2}}$. And thus an estimation of $\frac{1}{n}\sum_{q\in Q} f_{h1,h2}(q)$ provides an estimation of $\mathtt{d}_Q(h_1, h_2)$. From [63][Theorem 2.2] (see Lemma 5.1), the total sensitivity of F is $d + 1$. In particular, given the sensitivity score $\sigma(q)$ for each $q \in Q$, we can invoke [63][Lemma 2.1] to reach the following theorem.

Theorem 5.1. *Consider full rank $Q \subset \mathbb{R}^d$ and halfspaces $\mathcal{H}$ with $\varepsilon, \delta \in (0, 1)$. A σ-sensitive sampling $\tilde{Q}$ of (Q, F) of size $|\tilde{Q}| = \frac{d+1}{\delta,\varepsilon^2}$ results in a $(0, \varepsilon, \delta)$-coreset. And thus an $(0, \varepsilon, \delta)$-approximation so with probability at least $1 - \delta$, for each pair $h_1, h_2 \in \mathcal{H}$*

$$(1 - \varepsilon)\mathtt{d}_Q(h_1, h_2) \leq \mathtt{d}_{\tilde{Q},W}(h_1, h_2) \leq (1 + \varepsilon)\mathtt{d}_Q(h_1, h_2).$$

5.2.2 Sensitivity Computation and its Relationship with Leverage Score

In this section, we describe how to compute the sensitivity score $\sigma(x_i)$ for each $x_i \in Q$. To this end, we can invoke a theorem about vector norms by Langberg and Shulman [63]:

Lemma 5.1 (Theorem 2.2 in [63], expanding definitions)**.** *Suppose μ is a probability measure on a metric space X, and $V = \{v : X \mapsto \mathbb{R}\}$ is a real vector space of dimension κ. Let $F = \{f : X \mapsto [0, \infty) \mid \exists\, v \in V$ s.t. $f(x) = v(x)^2,\ \forall x \in X\}$, and $\{v^{(1)}, \cdots, v^{(\kappa)}\}$ be an orthonormal basis for V under the inner product $\langle u, v\rangle := \int_X u(x)v(x)\mathtt{d}\mu(x), \forall u, v \in V$. Then, $\sigma_{F,X,\mu}(x) = \sum_{i=1}^{\kappa} v^{(i)}(x)^2$ and $\mathfrak{S}(F) = \kappa$.*

We have already set $X = Q$ and $\mu = \frac{1}{n}$, and have defined V and F. To apply the above theorem need to define an orthonormal basis $\{v^{(1)}, v^{(2)}, \ldots, v^{(d+1)}\}$ for V. A straightforward basis (although not necessarily an orthonormal one) exists as $v^{(d+1)}(q) = v_{e^{(d+1)}}(q) =$

1 and $v^{(i)}(q) = v_{e^{(i)}}(q) = x_i$ for all $i \in [d]$ and $q = (x_1, \cdots, x_d) \in \mathbb{R}^d$, where $e^{(i)} = (0, \cdots, 0, 1, 0, \cdots, 0)$ is an indicator vector with all zeros except 1 in ith coordinate. That is the ith basis element $v^{(i)}$ is simply the ith coordinate of the input. Since Q is full rank, $\{v^{(1)}, \cdots, v^{(d+1)}\}$ is a basis of V.

We are now ready to state our theorem on computing sensitivity scores on a general (F, Q, μ), where we typically set $\mu = \frac{1}{n}$.

Theorem 5.2. *Suppose μ is a probability measure on a metric space $Q = \{q_1, \cdots, q_n\}$ such that $\mu(q_i) = p_i > 0$ for all $i \in [n]$, $V = \{v : Q \mapsto \mathbb{R}\}$ is a real vector space of dimension κ with a basis $\{v^{(1)}, \cdots, v^{(\kappa)}\}$, and $F = \{f : Q \mapsto [0, \infty) \mid \exists\, v \in V \text{ s.t. } f(q) = v(q)^2, \forall q \in Q\}$. If we introduce a $\kappa \times n$ matrix A whose ith column a_i is defined as: $a_i = (v^{(1)}(q_i)\sqrt{p_i}, \cdots, v^{(\kappa)}(q_i)\sqrt{p_i})^T$, then we have*

$$\sigma_{F,Q,\mu}(q_i) \cdot p_i = a_i^T (AA^T)^{-1} a_i, \quad \forall\, q_i \in Q. \tag{5.2}$$

Proof. Suppose the QR decomposition of A^T is $A^T = \tilde{Q}\tilde{R}$, where $\tilde{Q}$ is an $n \times \kappa$ orthogonal matrix ($\tilde{Q}^T\tilde{Q} = I$), and $\tilde{R}$ is an $n \times n$ upper triangular matrix. Since $\{v^{(1)}, \cdots, v^{(\kappa)}\}$ is a basis of V, the columns of A^T are linear independent, which implies the matrix $\tilde{R}$ is invertible. Using the fact that $\tilde{Q}^T\tilde{Q}$ is an identity matrix, we have

$$\begin{aligned} A^T(AA^T)^{-1}A =& \tilde{Q}\tilde{R}(\tilde{R}^T\tilde{Q}^T\tilde{Q}\tilde{R})^{-1}\tilde{R}^T\tilde{Q}^T = \tilde{Q}\tilde{R}(\tilde{R}^T\tilde{R})^{-1}\tilde{R}^T\tilde{Q}^T \\ =& \tilde{Q}\tilde{R}\tilde{R}^{-1}(\tilde{R}^T)^{-1}\tilde{R}^T\tilde{Q}^T = \tilde{Q}\tilde{Q}^T \end{aligned} \tag{5.3}$$

From Lemma 5.1, we have $\sigma_{F,Q,\mu}(q_i) = \sum_{j=1}^{\kappa}(\tilde{Q}_{i,j})^2$, which is the i-th entry on the diagonal of $\tilde{Q}\tilde{Q}^T$, so from (5.3), we obtain (5.2). □

This theorem not only shows how to compute the sensitivity of a point, but also gives the relationship between sensitivity and the leverage score.

Leverage score. Let $(\cdot)^+$ denotes the Moore-Penrose pseudoinverse of a matrix, so $(AA^T)^+ = (AA^T)^{-1}$ when AA^T is full rank. The *leverage score* [36] of the ith column a_i of matrix A is defined as: $\tau_i(A) := a_i^T(AA^T)^+ a_i$.

This definition is more specific and linear-algebraic than sensitivity. However, Theorem 5.2 shows that value $\sigma_{F,Q,\mu}(x_i) \cdot p_i$ is just the leverage score of the ith column of the matrix A. Compared to sensitivity, leverage scores have received more attention for scalable algorithm development and approximation [15, 25, 26, 35, 36, 74]

5.2.3 Estimate the Distance by Online Row Sampling

If the dimensionality is too high and the number of points is too large to be stored and processed in memory, we can apply online row sampling [26] to estimate $\mathtt{d}_Q$. Note that as more rows are witnessed the leverage score of older rows change. While other approaches (c.f. [25, 35, 74]) can obtain similar (and maybe slightly stronger) bounds, they rely on more complex procedures to manage these updating scores. The following Algorithm 5.1 by Cohen *et.al.* [26], on the other hand, simply samples columns as they come proportional to their estimated ridge leverage score [25]; thus it seems like the "right" approach.

Algorithm 5.1 ONLINE-SAMPLE(A, ε, δ)

Set $\lambda := \frac{\delta}{\varepsilon}$, $c := 8\log(\frac{d}{\varepsilon^2})$, and let $\widetilde{A}$ be empty (a $0 \times d$ matrix).
for rows $a_i \in A$ **do**
  Let $p_i := \min(c \cdot (1+\varepsilon) a_i^T (\widetilde{A}^T \widetilde{A} + \lambda I)^{-1} a_i, 1)$.
  With probability p_i, append row $a_i / \sqrt{p_i}$ to $\widetilde{A}$; otherwise do nothing.
return $\widetilde{A}$.

According to the Theorem 3 in [26], Algorithm 5.1 returns a matrix $\widetilde{A}$, with high probability, such that $(1-\varepsilon)A^TA - \delta I \preceq \widetilde{A}^T\widetilde{A} \preceq (1+\varepsilon)A^TA + \delta I$, and the number of rows in $\widetilde{A}$ is $O(d\log(d)\log(\varepsilon\|A\|_2^2/\delta)/\varepsilon^2)$. (Recall $A \preceq B$ means $x^TAx \leq x^TBx$ for every vector x.)

Given a set of points $Q = \{q_1, \cdots, q_n\} \subset \mathbb{R}^d$, where q_i has the coordinates $(x_{i,1}, \cdots, x_{i,d})$, we introduce an $n \times (d+1)$ matrix A_Q whose ith row a_i is defined as:

$$a_i = (x_{i,1}, \cdots, x_{i,d}, 1),$$

For any two hyperplanes h_1, h_2, they can be uniquely expressed by vectors $u^{(1)}, u^{(2)} \in \mathbb{U}^{d+1}$, and define $u = u^{(1)} - u^{(2)} \in \mathbb{R}^{d+1}$, then we have $\mathtt{d}_Q(h_1, h_2) = \frac{1}{\sqrt{n}}\|A_Q u\|$. So, if n is very large we can apply Algorithm 5.1 to efficiently sample rows from A_Q, and use $A_{\tilde{Q}}$ to estimate $\mathtt{d}_Q(h_1, h_2)$. From Theorem 3 in [26], we have the following result.

Theorem 5.3. *Suppose a set Q and matrix A_Q are defined as above. Let $A_{\tilde{Q}}$ = Online-Sample$(A_Q, \varepsilon, \delta)$ be the matrix returned by Algorithm 5.1. Then, with probability at least $1 - \frac{1}{d+1}$, for any two hyperplanes h_1, h_2 expressed by $u^{(1)}, u^{(2)} \in \mathbb{U}^{d+1}$, suppose $u_{h_1,h_2} = u^{(1)} - u^{(2)}$, we have*

$$\frac{1}{1+\varepsilon}\left(\frac{1}{n}\|A_{\tilde{Q}}u_{h_1,h_2}\|^2 - \frac{1}{n}\delta\|u_{h_1,h_2}\|^2\right)^{\frac{1}{2}} \leq \mathtt{d}_Q(h_1,h_2) \leq \frac{1}{1-\varepsilon}\left(\frac{1}{n}\|A_{\tilde{Q}}u_{h_1,h_2}\|^2 + \frac{1}{n}\delta\|u_{h_1,h_2}\|^2\right)^{\frac{1}{2}},$$

where $\|\cdot\|$ is the Euclidean norm, and with probability at least $1 - \frac{1}{d+1} - e^{-(d+1)}$ the number of rows in $A_{\tilde{Q}}$ is $O(d\log(d)\log(\varepsilon\|A_Q\|_2^2/\delta)/\varepsilon^2)$.

To make the above bound hold with arbitrarily high probability, we can use the standard median trick: run Algorithm 5.1 k times in parallel to obtain $A_{\tilde{Q}_1}, \cdots, A_{\tilde{Q}_k}$, then for any two hyperplanes h_1, h_2, we take the median of $\|A_{\tilde{Q}_1} u_{h_1,h_2}\|^2, \cdots, \|A_{\tilde{Q}_k} u_{h_1,h_2}\|^2$.

Remark 5.1. *Since $u_{h_1,h_2} = u^{(1)} - u^{(2)}$, we have*

$$\begin{aligned}\|u_{h_1,h_2}\|^2 =& (\|u^{(1)} - u^{(2)}\|)^2 \leq (\|u^{(1)}\| + \|u^{(2)}\|)^2 \leq 2(\|u^{(1)}\|^2 + \|u^{(2)}\|^2)\\ =& 2\big(2 + (u^{(1)}_{d+1})^2 + (u^{(2)}_{d+1})^2\big) = 4 + 2\mathtt{d}^2(\mathbf{0}, h_1) + 2\mathtt{d}^2(\mathbf{0}, h_2),\end{aligned}$$

where $\mathtt{d}(\mathbf{0}, h)$ is the distance from a choice of origin $\mathbf{0}$ to h. If we assume that any hyperplanes we consider must pass within a distance Δ to the choice of origin, then let $\Delta' = 4(1+\Delta^2)$ and $\|u_{h_1,h_2}\|^2 \leq \Delta'$. Now $\mathtt{d}_{\tilde{Q},W}(h_1,h_2))^2 = \frac{1}{n}\|A_{\tilde{Q}} u_{h_1,h_2}\|^2$ where $\tilde{Q}$ is the set of points corresponding to rows in $A_{\tilde{Q}}$, and the weighting W is defined so $w_i = |\tilde{Q}|/n$. Then the conclusion of Theorem 5.3 can be rewritten as

$$\frac{1}{1+\varepsilon}\big(\mathtt{d}_{\tilde{Q},W}(h_1,h_2)^2 - \frac{\Delta'\delta}{n}\big)^{\frac{1}{2}} \leq \mathtt{d}_Q(h_1,h_2) \leq \frac{1}{1-\varepsilon}\big(\mathtt{d}_{\tilde{Q},W}(h_1,h_2)^2 + \frac{\Delta'\delta}{n}\big)^{\frac{1}{2}},$$

which means $\mathtt{d}_Q(h_1,h_2)$ can be estimated by $\mathtt{d}_{\tilde{Q},W}(h_1,h_2)$ and the bound Δ on the distance to the origin. Recall the distance and the bound in Theorem 5.3 is invariant to the choice of $\mathbf{0}$, so for this interpretation it can always be considered so Δ is small.

5.2.4 A Strong $O(0,\varepsilon,\delta)$-Approximation for Q over $\mathcal{H}$.

Now, we use the framework in Braverman *et.al.* [16] to construct a strong $O(0,\varepsilon,\delta)$-approximation for Q over $\mathcal{H}$. In the remaining part of this subsection, we assume Q is a set (not a multiset), each $q \in Q$ has a weight $w(q) \in (0,1]$, and $\sum_{q\in Q} w(q) = 1$. Recall that for a range space $(Q,\mathcal{R})$ the shattering dimension $\mathfrak{s} = \dim(Q,\mathcal{R})$ is the smallest integer $\mathfrak{s}$ so that $|\{S \cap R \mid R \in \mathcal{R}\}| \leq |S|^{\mathfrak{s}}$ for all $S \subset Q$. We introduce ranges $\mathcal{X}$ where each range $X_{h_1,h_2,\eta} \in \mathcal{X}$ is defined by two halfspaces $h_1, h_2 \in \mathcal{H}$ and a threshold $\eta > 0$. This is defined with respect to Q and a weighting $w : Q \to \mathbb{R}_+$, specifically

$$X_{h_1,h_2,\eta} = \{q \in Q \mid w(q) f_{h_1,h_2}(q) \leq \eta\}.$$

Next we use the sensitivity $\sigma : Q \to \mathbb{R}_+$ to define an adjusted range space $(Q, \mathcal{X}')$ with adjusted weights $w'(q) = \frac{\sigma(q)}{d+1} w(q)$ and adjusted ranges $X'_{h_1,h_2,\eta} \in \mathcal{X}'$ defined using $g_{h_1,h_2}(q) = \frac{1}{\sigma(q)} \frac{f_{h_1,h_2}(q)}{\bar{f}_{h_1,h_2}}$ as

$$X'_{h_1,h_2,\eta} = \{q \in Q \mid w'(q) g_{h_1,h_2}(q) \leq \eta\}.$$

Recall that $\bar{f}_{h_1,h_2} = \sum_{q \in Q} w(q) f_{h_1,h_2}(q)$. To apply [16][Theorem 5.5] we only need to bound the shattering dimension of the adjusted range space $(Q, \mathcal{X}')$.

Here is a lemma about the computation of the dimension of a range space, which is useful in bounding the dimension of a query space.

Lemma 5.2. *Suppose $Q \subset \mathbb{R}^d, X_1 \subset \mathbb{R}^{d_1}, X_2 \subset \mathbb{R}^{d_2}$, and $\mathcal{R}_1 = \{\{q \in Q | \, g_1(q,x) \leq 0\} | \, x \in X_1\}$, $\mathcal{R}_2 = \{\{q \in \mathbb{R}^2 | \, g_2(q,x) \leq 0\} | \, x \in X_2\}$ where g_1, g_2 can be any fixed real functions. Define $\mathcal{R}_3 = \{\{q \in \mathbb{R}^2 | \, g_1(q,x_1) \leq 0\} \cap \{q \in \mathbb{R}^2 | \, g_2(q,x_2) \leq 0\} | \, x_1 \in X_1, x_2 \in X_2\}$, $\mathcal{R}_4 = \{\{q \in \mathbb{R}^2 | \, g_1(q,x_1) \leq 0\} \cup \{q \in \mathbb{R}^2 | \, g_2(q,x_2) \leq 0\} | \, x_1 \in X_1, x_2 \in X_2\}$. If $dim(\mathbb{R}^2, \mathcal{R}_1) = s_1$ and $dim(\mathbb{R}^2, \mathcal{R}_2) = s_2$, then $dim(\mathbb{R}^2, \mathcal{R}_3) \leq s_1 + s_2$ and $dim(\mathbb{R}^2, \mathcal{R}_4) \leq s_1 + s_2$.*

Proof. Suppose $G \subset \mathbb{R}^2$ and $|G| \leq \infty$, then we have

$$\{G \cap R | \, R \in \mathcal{R}_3\} = \{(G \cap R_1) \cap (G \cap R_2) | \, R_1 \in \mathcal{R}_1, R_2 \in \mathcal{R}_2\}. \tag{5.4}$$

So, we have

$$\begin{aligned} |\{G \cap R | \, R \in \mathcal{R}_3\}| &= |\{(G \cap R_1) \cap (G \cap R_2) | \, R_1 \in \mathcal{R}_1, R_2 \in \mathcal{R}_2\}| \\ \leq &|\{G \cap R_1 | \, R_1 \in \mathcal{R}_1\}| \times |\{G \cap R_2 | \, R_2 \in \mathcal{R}_2\}| \leq |G|^{s_1} |G|^{s_2} = |G|^{s_1+s_2}. \end{aligned} \tag{5.5}$$

which implies $\dim(\mathbb{R}^2, \mathcal{R}_3) \leq s_1 + s_2$, and similarly we have $\dim(\mathbb{R}^2, \mathcal{R}_4) \leq s_1 + s_2$. □

Now, we can bound the shattering dimension of the adjusted range space $(Q, \mathcal{X}')$.

Lemma 5.3. *The shattering dimension of adjusted range space $(Q, \mathcal{X}')$ is bounded by $O(d)$.*

Proof. We start by rewriting any element $X'_{h_1,h_2,\eta}$ of the adjusted range space as

$$\begin{aligned} X'_{h_1,h_2,\eta} &= \{q \in Q \mid w'(q) g_{h_1,h_2}(x) \leq \eta\} \\ &= \{q \in Q \mid w(q) f_{h_1,h_2}(q) \leq \eta(d+1) \bar{f}_{h_1,h_2}\} \\ &= \{q \in Q \mid \sqrt{w(q)} \big(\textstyle\sum_{i=1}^d u_i x_i + u_{d+1} \big) \big) \leq \big(\eta(d+1) \bar{f}_{h_1,h_2} \big)^{\frac{1}{2}}\} \\ &\quad \cap \{q \in Q \mid -\sqrt{w(q)} \big(\textstyle\sum_{i=1}^d u_i x_i + u_{d+1} \big) \big) \leq \big(\eta(d+1) \bar{f}_{h_1,h_2} \big)^{\frac{1}{2}}\}, \end{aligned}$$

where $(x_1, \cdots, x_d)$ is the coordinates of $q \in Q$. This means each set $X'_{h_1,h_2,\eta} \in \mathcal{X}'$ can be decomposed as the intersection of sets in two ranges over Q from:

$$\mathcal{R}_1 = \Big\{ \{q \in Q \mid \sqrt{w(q)}\big(\textstyle\sum_{i=1}^d u_i x_i + u_{d+1}\big)\big) \leq \big(\eta(d+1)\bar{f}_{h_1,h_2}\big)^{\frac{1}{2}} \} \mid h_1, h_2 \in \mathcal{H}, \eta \geq 0 \Big\},$$
$$\mathcal{R}_2 = \Big\{ \{q \in Q \mid -\sqrt{w(q)}\big(\textstyle\sum_{i=1}^d u_i x_i + u_{d+1}\big)\big) \leq \big(\eta(d+1)\bar{f}_{h_1,h_2}\big)^{\frac{1}{2}} \} \mid h_1, h_2 \in \mathcal{H}, \eta \geq 0 \Big\}.$$

By Lemma 5.2, we only need to bound the dimension of each associated range space $(Q, \mathcal{R}_1)$ and $(Q, \mathcal{R}_2)$. We introduce new variables $c_0 \in \mathbb{R}, z = (z_1, \cdots, z_{d+1}), c = (c_1, \cdots, c_{d+1}) \in \mathbb{R}^{d+1}$:

$$z_i = \sqrt{w(q)} x_i \text{ for } i \in [d], \;\; z_{d+1} = \sqrt{w(q)},$$
$$c_i = u_i \text{ for } i \in [d+1], \;\; c_0 = -\big(r(d+1)\bar{f}_{h_1,h_2}\big)^{\frac{1}{2}}.$$

Since Q is a fixed set, we know z only depends on q, and c_0, c only depend on h_1, h_2 and η. By introducing new variables we construct an injective map $\varphi : Q \mapsto \mathbb{R}^{d+1}$, s.t. $\varphi(q) = z$. So, there is also an injective map from $\mathcal{R}_1$ to $\big\{\{z \in \varphi(Q) \mid c_0 + \langle z, c\rangle \leq 0\} \mid c_0 \in \mathbb{R}, c \in \mathbb{R}^{d+1}\big\}$. Since the shattering dimension of the range space $(\mathbb{R}^{d+1}, \mathcal{H}^{d+1})$, where $\mathcal{H}^{d+1} = \{h \text{ is a halfspace in } \mathbb{R}^{d+1}\}$, is $O(d)$, we have $\dim(Q, \mathcal{R}_1) = O(d)$, and similarly $\dim(Q, \mathcal{R}_2) = O(d)$. Thus, we obtain an $O(d)$ bound for the shattering dimension of $(Q, \mathcal{X})$. □

From Lemma 5.3 and [16][Theorem 5.5] we directly obtain a strong $O(0, \varepsilon, \delta)$-approximation for Q over $\mathcal{H}$.

Theorem 5.4. *Consider full rank $Q \subset \mathbb{R}^d$ and halfspaces $\mathcal{H}$ with $\varepsilon, \delta \in (0,1)$. A σ-sensitive sampling $\tilde{Q}$ of (Q, F) of size $|\tilde{Q}| = O(\frac{d}{\varepsilon^2}(d \log d + \log \frac{1}{\delta}))$ results in a strong $(0, \varepsilon, \delta)$-coreset. And thus a strong $(0, \varepsilon, \delta)$-approximation so with probability at least $1 - \delta$, for all $h_1, h_2 \in \mathcal{H}$*

$$(1-\varepsilon)\mathtt{d}_Q(h_1, h_2) \leq \mathtt{d}_{\tilde{Q},W}(h_1, h_2) \leq (1+\varepsilon)\mathtt{d}_Q(h_1, h_2).$$

5.3 Distance Between Two Geometric Objects

In this section, we mildly restrict $\mathtt{d}_Q$ to the distance between any two geometric objects, in particularly bounded closed sets. Let $\mathcal{S} = \{S \subset \mathbb{R}^d \mid S \text{ is a bounded closed set}\}$ be the space of objects $\mathcal{J}$ we consider.

As before define $v_i(S) = \inf_{p \in S} \|p - q_i\|$, and then for $S_1, S_2 \in \mathcal{S}$ define $f_{S_1,S_2}(q_i) = (v_i(S_1) - v_i(S_2))^2$. The associated function space is $F(\mathcal{S}) = \{f_{S_1,S_2} \mid S_1, S_2 \in \mathcal{S}\}$. Setting

$\mu(q) = \frac{1}{n}$ for all $q \in Q$, then $(\mathtt{d}_Q(S_1, S_2))^2 = \bar{f}_{S_1,S_2} := \sum_{i=1}^{n} \mu(q_i) f_{S_1,S_2}(q_i)$. Using sensitivity sampling to estimate $\mathtt{d}_Q(S_1, S_2)$ requires a bound on the total sensitivity of $F(\mathcal{S})$.

In this section we show that while unfortunately the total sensitivity $\mathfrak{S}(F(\mathcal{S}))$ is unbounded in general, it can be tied closely to the ratio L/ρ between the diameter of the domain L, and the minimum allowed $\mathtt{d}_Q$ distance between objects ρ. In particular, it can be at least proportional to this, and in $\mathbb{R}^2$ in most cases (e.g., for near-uniform Q) is at most proportional to L/ρ or not much larger for any Q.

5.3.1 Lower Bound on Total Sensitivity

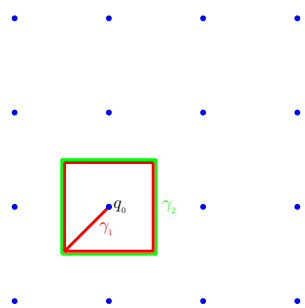

Figure 5.1: Q is the set of blue points, γ_1 is the red curve, γ_2 is the green curve, and they coincide with each other on the boundary of the square.

Suppose Q is a set of n points in $\mathbb{R}^2$ and no two points are at the same location, then for any $q_0 \in Q$ we can draw two curves γ_1, γ_2 as shown in Figure 5.1, where γ_1 is composed by five line segments and γ_2 is composed by four line segments. The four line segments of the γ_2 forms a square, on its boundary γ_1 and γ_2 coincide with each other, and inside this square, q_0 is the endpoint of γ_1. We can make this square small enough, such that all points $q \neq q_0$ are outside this square. So, we have $\mathrm{dist}(q_0, \gamma_1) = 0$ and $\mathrm{dist}(q_0, \gamma_2) \neq 0$, and $\mathrm{dist}(q, \gamma_1) = \mathrm{dist}(q, \gamma_2) = 0$ for all $q \neq q_0$. Thus, we have $f_{\gamma_1,\gamma_2}(q_0) > 0$ and $f_{\gamma_1,\gamma_2}(q) = 0$ for all $q \neq q_0$, which implies

$$\sigma_{F(\mathcal{S}),Q,\mu}(q_0) \geq \frac{f_{\gamma_1,\gamma_2}(q_0)}{\bar{f}_{\gamma_1,\gamma_2}} = \frac{f_{\gamma_1,\gamma_2}(q_0)}{\frac{1}{n}\sum_{q \in Q} f_{\gamma_1,\gamma_2}(q)} = \frac{n f_{\gamma_1,\gamma_2}(q_0)}{f_{\gamma_1,\gamma_2}(q_0)} = n.$$

Since this construction of two curves γ_1, γ_2 can be repeated around any point $q \in Q$,

$$\mathfrak{S}(F(\mathcal{S})) = \sum_{q \in Q} \mu(q) \sigma_{F(\mathcal{S}),Q,\mu}(q) \geq \sum_{q \in Q} \frac{1}{n} n = n.$$

We can refine this bound by introducing two parameters L, ρ for $\mathcal{S}$. Given $L > \rho > 0$ and a set $Q \subset \mathbb{R}^d$ of n points, we define $\mathcal{S}(L) = \{S \in \mathcal{S} \mid S \subset [0, L]^d\}$ and $F(\mathcal{S}(L), \rho) = \{f_{S_1,S_2} \in F(\mathcal{S}) \mid S_1, S_2 \in \mathcal{S}(L),\ \mathtt{d}_Q(S_1, S_2) \geq \rho\}$. The following lemma gives a lower bound for the total sensitivity of $F(\mathcal{S}(L), \rho)$ in the case $d = 2$, which directly holds for larger d.

Lemma 5.4. *Suppose $d = 2$, then can construct a set $Q \subset [0, L]^d$ such that $\mathfrak{S}(F(\mathcal{S}(L), \rho)) = \Omega(\frac{L}{\rho})$.*

Proof. We uniformly partition $[0, L]^2$ into n grid cells, such that $C_1 \frac{L}{\rho} \leq n \leq C_2 \frac{L}{\rho}$ for constants $C_1, C_2 \in (0, 1)$. The side length of each grid is $\eta = \frac{L}{\sqrt{n}}$. We take Q as the n grid points, and for each point $q \in Q$ we can choose two curves γ_1 and γ_2 (similar to curves in Figure 5.1) such that $\mathrm{dist}(q, \gamma_1) = 0$, $\mathrm{dist}(q, \gamma_2) \geq C_2\eta$, and $\mathrm{dist}(q', \gamma_1) = \mathrm{dist}(q', \gamma_2) = 0$ for all $q' \in Q \setminus \{q\}$. Thus, we have $\mathtt{d}_Q(\gamma_1, \gamma_2) \geq C_2 \frac{\eta}{\sqrt{n}} = C_2 \frac{L}{n} \geq \rho$. So, $f_{\gamma_1,\gamma_2} \in F(\mathcal{S}(L), \rho))$ and we have $\sigma(q) \geq n$ for all $q \in Q$ and $\mathfrak{S}(F(\mathcal{S}(L), \rho)) \geq n \geq C_1 \frac{L}{\rho}$, which implies $\mathfrak{S}(F(\mathcal{S}(L), \rho)) = \Omega(\frac{L}{\rho})$. □

5.3.2 Upper Bound on the Total Sensitivity

A simple upper bound of $\mathfrak{S}(F(\mathcal{S}(L), \rho)$ is $O(\frac{L^2}{\rho^2})$ follows from the L/ρ constraint. The sensitivity of each point $q \in Q$ is defined as $\sup_{f_{S_1,S_2} \in F(\mathcal{S}(L),\rho)} \frac{f_{S_1,S_2}(q)}{\bar{f}_{S_1,S_2}}$, where $f_{S_1,S_2}(q) = O(L^2)$ for all $S_1, S_2 \in \mathcal{S}(L)$ and $q \in Q \subset [0, L]^d$, and the denominator $\bar{f}_{S_1,S_2} \geq \rho^2$ by assumption for all $f_{S_1,S_2} \in F(\mathcal{S}(L), \rho)$. Hence, the sensitivity of each point in Q is $O(\frac{L^2}{\rho^2})$, and thus their average, the total sensitivity is $O(\frac{L^2}{\rho^2})$. In this section we will improve and refine this bound.

We introduce two variables only depends on $Q = \{q_1, \cdots, q_n\} \subset [0, L]^d$:

$$C_q := \max_{0<r\leq L} \frac{r^d}{L^d} \frac{n}{|Q \cap B_\infty(q, r)|} \quad \text{for } q \in Q, \text{ and } C_Q := \frac{1}{n} \sum_{q \in Q} C_q^{\frac{2}{2+d}}. \tag{5.6}$$

where $B_\infty(q, r) := \{x \in \mathbb{R}^d \mid \|x - q\|_\infty \leq r\}$. Intuitively, $\frac{|Q \cap B_\infty(q,r)|}{r^d}$ is proportional to the point density in region $B_\infty(q, r)$, and the value of $\frac{r^d}{L^d} \frac{n}{|Q \cap B_\infty(q,r)|}$ can be maximized, when the region $B_\infty(q, r)$ has smallest point density, which means r should be as large as possible but the number of points contained in $B_\infty(q, r)$ should be as small as possible. A trivial bound of C_q is n, but if we make $C_{q_0} = n$ for one point q_0, then it implies the value of C_q for other points will be small, so for C_Q it is possible to obtain a bound better than $n^{\frac{2}{d+2}}$.

Importantly, these quantities C_q and C_Q will be directly related to the sensitivity of a single point $\sigma(q)$ and the total sensitivity of the point set $\mathfrak{S}_Q$, respectively. We formalize this connection in the next lemma, which for instance implies that for d constant then $\mathfrak{S}_Q = O(C_Q \cdot (L/\rho)^{\frac{2d}{2+d}})$.

Lemma 5.5. *For function family $F(\mathcal{S}(L), \rho)$ the sensitivity for any $q \in Q \in [0, L]^d$ is bounded*

$$\sigma(q) \leq C_d C_q^{\frac{2}{2+d}} \left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}},$$

where $C_d = 4^{\frac{2}{2+d}} (8\sqrt{d})^{\frac{2d}{2+d}}$ and C_q given by (5.6).

Proof. Recall $\sigma(q) = \sup_{f_{S_1,S_2} \in F(\mathcal{S}(L),\rho)} \frac{f_{S_1,S_2}(q)}{\frac{1}{n}\sum_{q' \in Q} f_{S_1,S_2}(q')}$. For any fixed $q \in Q$, for now suppose $f_{S_1,S_2} \in F(\mathcal{S}(L), \rho)$ satisfies this supremum $\sigma(q) = \frac{f_{S_1,S_2}(q)}{\frac{1}{n}\sum_{q' \in Q} f_{S_1,S_2}(q')}$. We define $\text{dist}(q, S) = \inf_{p \in S} \|q - p\|$ (so for $q_i \in Q$ then $\text{dist}(q_i, S) = v_i(S)$), and then use the parameter $M := |\text{dist}(q, S_1) - \text{dist}(q, S_2)|$, where $M^2 = f_{S_1,S_2}(q)$. If $M = 0$, then obviously $f_{S_1,S_2}(q) = M^2 = 0$, and $\sigma(q) = 0$. So, without loss of generality, we assume $M > 0$ and $\text{dist}(q, S_1) = \tau$ and $\text{dist}(q, S_2) = \tau + M$. We first prove $\sigma(q) \leq C_d C_q \frac{L^d}{M^d}$. There are two cases for the relationship between τ and M, as shown in Figure 5.2.

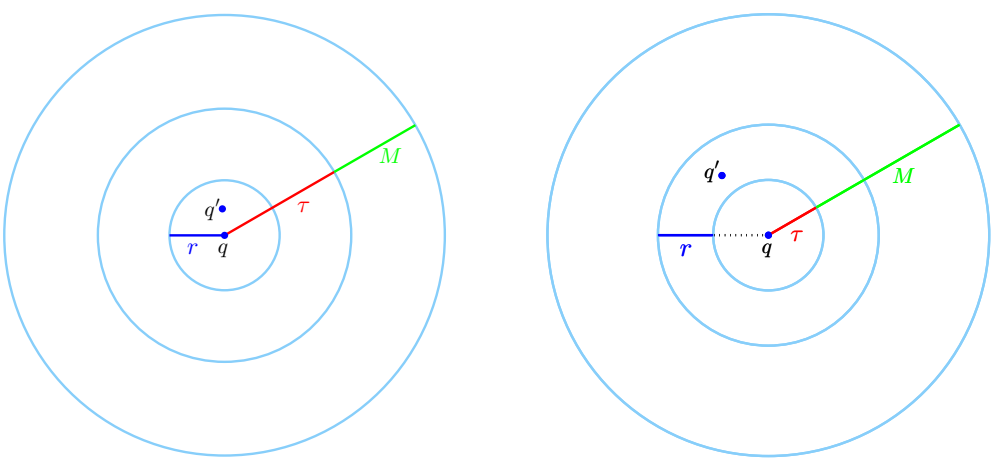

Figure 5.2: Left: Case 1, $r = \frac{M}{8} \leq \tau$, and $q' \in B(q, r)$. Right: Case 2, $r = \frac{M}{8} > \tau$, and $q' \in B(q, \tau + r)$.

Case 1: $\tau \geq \frac{M}{8}$**.** For any $q' \in B(q, \frac{M}{8}) := \{q' \in \mathbb{R}^d \mid \|q' - q_i\| \leq \frac{M}{8}\}$, we have $\tau + M = \text{dist}(q, S_2) \leq \text{dist}(q, q') + \text{dist}(q', S_2) \leq \frac{M}{8} + \text{dist}(q', S_2)$, which implies for all $q' \in B(q, \frac{M}{8})$

$$\text{dist}(q', S_2) \geq \tau + M - \frac{M}{8} = \tau + \frac{7}{8}M.$$

Similarly $\mathrm{dist}(q', S_1) \le \mathrm{dist}(q', q) + \mathrm{dist}(q, S_1) \le \frac{M}{8} + \tau$ for all $q' \in B(q, \frac{M}{8})$. Thus for all $q' \in B(q, \frac{M}{8})$

$$|\mathrm{dist}(q', S_2) - \mathrm{dist}(q', S_1)| \ge \mathrm{dist}(q', S_2) - \mathrm{dist}(q', S_1) \ge \tau + \frac{7}{8}M - (\tau + \frac{M}{8}) = \frac{3}{4}M.$$

Case 2: $0 \le \tau < \frac{M}{8}$**.** For any $q' \in B(q, \tau + \frac{M}{8}) := \{q' \in \mathbb{R}^d \mid \mathrm{dist}(q', q) \le \tau + \frac{M}{8}\}$, we have $\tau + M = \mathrm{dist}(q', S_2) \le \mathrm{dist}(q, q') + \mathrm{dist}(q', S_2) \le \tau + \frac{M}{8} + \mathrm{dist}(q', S_2)$, which implies for all $q' \in B(q, \tau + \frac{M}{8})$

$$\mathrm{dist}(q', S_2) \ge \frac{7}{8}M.$$

Combined with $\tau < \frac{M}{8}$ and $\mathrm{dist}(q', S_1) \le \mathrm{dist}(q', q) + \mathrm{dist}(q, S_1) \le \tau + \frac{M}{8} + \tau = \frac{M}{8} + \frac{M}{8} + \frac{M}{8} \le \frac{3}{8}M$ for all $q' \in B(q, \tau + \frac{M}{8})$, we have

$$|\mathrm{dist}(q', S_2) - \mathrm{dist}(q', S_1)| \ge \mathrm{dist}(q', S_2) - \mathrm{dist}(q', S_1) \ge \frac{7}{8}M - \frac{3}{8}M = \frac{M}{2}.$$

Combining these two cases on τ, for all $q' \in B(q, \frac{M}{8})$

$$|\mathrm{dist}(q', S_2) - \mathrm{dist}(q', S_1)| \ge \frac{M}{2}.$$

Then since $B_\infty(q, \frac{r}{\sqrt{d}}) \subset B(q, r)$ for all $r \ge 0$, from

$$C_q = \max_{0<r\le L} \frac{r^d}{L^d} \frac{n}{|Q \cap B_\infty(q, r)|} \ge (\frac{1}{8\sqrt{d}})^d \frac{M^d}{L^d} \frac{n}{|Q \cap B_\infty(q, \frac{M}{8\sqrt{d}})|},$$

we can bound the denominator in $\sigma(q)$ as

$$\begin{aligned}\frac{1}{n}\sum_{q'\in Q} f_{S_1,S_2}(q') &\ge \frac{1}{n}\sum_{q'\in Q\cap B_\infty(q,\frac{M}{8\sqrt{d}})} f_{S_1,S_2}(q') = \frac{1}{n}\sum_{q'\in Q\cap B_\infty(q,\frac{M}{8\sqrt{d}})} (\mathrm{dist}(q', S_1) - \mathrm{dist}(q', S_2))^2 \\ &\ge \frac{1}{4}\frac{1}{n}M^2\Big|Q \cap B_\infty(q, \frac{M}{8\sqrt{d}})\Big| \ge \frac{1}{4}(\frac{1}{8\sqrt{d}})^d \frac{M^2}{C_q}\frac{M^d}{L^d} = \frac{1}{4}(\frac{1}{8\sqrt{d}})^d \frac{1}{C_q}\frac{M^{2+d}}{L^d},\end{aligned}$$

which implies

$$\sigma(q) = \frac{M^2}{\frac{1}{n}\sum_{q'\in Q} f_{S_1,S_2}(q')} \le 4(8\sqrt{d})^d M^2 C_q \frac{L^d}{M^{2+d}} = 4(8\sqrt{d})^d C_q \frac{L^d}{M^d}.$$

Combining this with $\sigma(q) \le \frac{M^2}{\rho^2}$, we have $\sigma(q) \le \min\left(\frac{M^2}{\rho^2}, 4(8\sqrt{d})^d C_q \frac{L^d}{M^d}\right)$. If $M^{2+d} \le 4(8\sqrt{d})^d C_q \rho^2 L^d$, then $\frac{M^2}{\rho^2} \le 4(8\sqrt{d})^d C_q \frac{L^d}{M^d}$, which means $\sigma(q) \le \min\left(\frac{M^2}{\rho^2}, 4(8\sqrt{d})^d C_q \frac{L^d}{M^d}\right) = \frac{M^2}{\rho^2} \le 4^{\frac{2}{2+d}}(8\sqrt{d})^{\frac{2d}{2+d}} C_q^{\frac{2}{2+d}} \left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}}$. If $M^{2+d} \ge 4(8\sqrt{d})^d C_q \rho^2 L^d$, then $4(8\sqrt{d})^d C_q \frac{L^d}{M^d} \le \frac{M^2}{\rho^2}$, so we also have $\sigma(q) \le \min\left(\frac{M^2}{\rho^2}, 4(8\sqrt{d})^d C_q \frac{L^d}{M^d}\right) = 4(8\sqrt{d})^d C_q \frac{L^d}{M^d} \le 4^{\frac{2}{2+d}}(8\sqrt{d})^{\frac{2d}{2+d}} C_q^{\frac{2}{2+d}} \left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}}$.

□

Hence, to bound the total sensitivity of $F(\mathcal{S}(L), \rho)$, we need a bound of $C_Q = \frac{1}{n}\sum_{q\in Q} C_q^{\frac{2}{2+d}}$.

Lemma 5.6. *Suppose $Q \subset [0, L]^d$ of size n, $\eta = \min_{q,q'\in Q,\, q\neq q'} \|q - q'\|_\infty$, and C_Q is given by* (5.6)*. Then we have*

$$C_Q \leq C_d \min\Big(\big(\log_2 \frac{L}{\eta}\big)^{\frac{2}{2+d}}, \big(\frac{1}{d}\log_2 n\big)^{\frac{2}{2+d}}\Big),$$

where $C_d = 2^{d+1}$.

Proof. We define $\widetilde{C}_Q := \frac{1}{n}\sum_{q\in Q} C_q$, and using Hölder inequality we have

$$C_Q = \frac{1}{n}\sum_{q\in Q} C_q^{\frac{2}{2+d}} \leq \frac{1}{n}\Big(\sum_{q\in Q} C_q\Big)^{\frac{2}{2+d}} n^{\frac{d}{2+d}} = \Big(\frac{1}{n}\sum_{q\in Q} C_q\Big)^{\frac{2}{2+d}} = (\widetilde{C}_Q)^{\frac{2}{2+d}}.$$

So, we only need to bound $\widetilde{C}_Q$.

We define $r_q := \arg\max_{0<r\leq L} \frac{r^d}{L^d}\frac{n}{|Q\cap B_\infty(q,r)|}$ for all $q \in Q$, $Q_i := \{q \in Q \mid \frac{L}{2^{i+1}} < r_q \leq \frac{L}{2^i}\}$, and $A := \{i \geq 0 \mid i \text{ is an integer and } |Q_i| > 0\}$.

For any fixed $i \in A$, we use $l_i := \frac{L}{2^{i+1}}$ as the side length of grid cell to partition the region $[0, L]^d$ into $s_i = (\frac{L}{l_i})^d = 2^{(i+1)d}$ grid cells: $\Omega_1. \cdots, \Omega_{s_i}$ where each Ω_j is a closed set, and define $Q_{i,j} := Q_i \cap \Omega_j$. Then, $|Q_i \cap \bar{B}_\infty(q, l_i)| \geq |Q_{i,j}|$ for all $q \in Q_{i,j}$ where $\bar{B}_\infty(q, l_i) := \{q' \in \mathbb{R}^d \mid \|q' - q\|_\infty \leq l_i\}$, and we have

$$\begin{aligned}
\sum_{q\in Q_i} \frac{r_q^d}{L^d}\frac{1}{|Q_i\cap B_\infty(q,r_q)|} &\leq \sum_{q\in Q}\frac{L^d}{2^{id}L^d}\frac{1}{|Q_i\cap B_\infty(q,r_q)|} \leq \frac{1}{2^{id}}\sum_{q\in Q_i}\frac{1}{|Q_i\cap \bar{B}_\infty(q,l_i)|}\\
&\leq \frac{1}{2^{id}}\sum_{j\in[s_i],|Q_{i,j}|>0}\sum_{q\in Q_{i,j}}\frac{1}{|Q_i\cap \bar{B}_\infty(q,l_i)|} \leq \frac{1}{2^{id}}\sum_{j\in[s_i],|Q_{i,j}|>0}\sum_{q\in Q_{i,j}}\frac{1}{|Q_{i,j}|}\\
&= \frac{1}{2^{id}}\sum_{j\in[s_i],|Q_{i,j}|>0}\frac{|Q_{i,j}|}{|Q_{i,j}|} \leq \frac{s_i}{2^{id}} = \frac{2^{(i+1)d}}{2^{id}} = 2^d.
\end{aligned}$$

Then using the definitions of $\widetilde{C}_Q$ and r_q we have

$$\begin{aligned}
\widetilde{C}_Q &= \sum_{q\in Q}\max_{0<r\leq L}\frac{r^d}{L^d}\frac{1}{|Q\cap B_\infty(q,r)|} = \sum_{q\in Q}\frac{r_q^d}{L^d}\frac{1}{|Q\cap B_\infty(q,r_q)|} = \sum_{i\in A}\sum_{q\in Q_i}\frac{r_q^d}{L^d}\frac{1}{|Q\cap B_\infty(q,r)|}\\
&\leq \sum_{i\in A}\sum_{q\in Q_i}\frac{r_q^d}{L^d}\frac{1}{|Q_i\cap B_\infty(q,r)|} \leq \sum_{i\in A} 2^d = 2^d|A|.
\end{aligned}$$

We assert $r_q \geq Ln^{-\frac{1}{d}}$ for all $q \in Q$. This is because for any $r \in (0, Ln^{-\frac{1}{d}})$ we have

$$\frac{r^d}{L^d}\frac{n}{|Q\cap B_\infty(q,r)|} \leq \frac{L^d}{nL^d}\frac{n}{1} = 1 \leq \frac{L^d}{L^d}\frac{n}{|Q\cap B_\infty(q,L)|},$$

which implies the optimal $r_q \in [Ln^{-\frac{1}{d}}, L]$. Moreover, since $r_q \geq \min_{q'\in Q,\, q'\neq q}\|q - q'\|_\infty \geq \eta$, we have $r_q \geq \max(Ln^{-\frac{1}{d}}, \eta)$ for all $q \in Q$. If $i > \min\big(\log_2 \frac{L}{\eta}, \frac{1}{d}\log_2 n\big)$, then $\frac{L}{2^i} <$

$\max(Ln^{-\frac{1}{d}}, \eta) \leq r_q$, and from the definition of Q_i and A we know $i \notin A$, which implies $|A| \leq 1 + \min\left(\log_2 \frac{L}{\eta}, \frac{1}{d}\log_2 n\right)$. Hence we obtain $\widetilde{C}_Q \leq 2^{d+1}\min\left(\log_2 \frac{L}{\eta}, \frac{1}{d}\log_2 n\right)$ and using $C_Q = (\widetilde{C}_Q)^{\frac{2}{2+d}}$ we prove the lemma. □

Since $f_{S_1,S_2} \in F(\mathcal{S}(L), \rho)$, we know $f_{S_1,S_2}(q) \leq dL^2$ for all $q \in Q$ and $\frac{1}{n}\sum_{q' \in Q} f_{S_1,S_2}(q') \geq \rho^2$, so $\sigma(q) \leq \frac{dL^2}{\rho^2}$ for all $q \in Q$. Thus, we can expand $\frac{1}{|Q|}\sum_{q \in Q}\sigma(q)$ using Lemma 5.5 and factor out C_Q using Lemma 5.6 to immediately obtain the following theorem about the total sensitivity of $F(\mathcal{S}(L), \rho)$.

Theorem 5.5. *Suppose $L > \rho > 0$, $Q = \{q_1, \cdots, q_n\} \subset [0, L]^d$ and $\eta = \min_{q,q' \in Q,\, q \neq q'} \|q - q'\|_\infty$. Then, we have*

$$\mathfrak{S}(F(\mathcal{S}(L), \rho)) \leq \mathfrak{S}_Q = O\left(\left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}} \min\left(\log\frac{L}{\eta}, \log n, \left(\frac{L}{\rho}\right)^2\right)^{\frac{2}{2+d}}\right).$$

From Lemma 5.5 and Theorem 5.5, using [63][Lemma 2.1] we can obtain the following theorem.

Theorem 5.6. *Let $L > \rho > 0$, $Q = \{q_1, \cdots, q_n\} \subset [0, L]^d$, $S_1, S_2 \in \mathcal{S}(L)$ and $\mathtt{d}_Q(S_1, S_2) \geq \rho$. Suppose $\sigma(q)$ and $\mathfrak{S}_Q$ are defined in Lemma 5.5 and Theorem 5.5 respectively. Then for $\delta, \varepsilon \in (0, 1)$ a σ-sensitive sampling of size $N \geq \frac{\mathfrak{S}_Q}{\delta\varepsilon^2}$ provides $\tilde{Q}$, a $(\rho, \varepsilon, \delta)$-coreset; that is with probability at least $1 - \delta$, we have*

$$(1 - \varepsilon)\mathtt{d}_Q(S_1, S_2) \leq \mathtt{d}_{\tilde{Q},W}(S_1, S_2) \leq (1 + \varepsilon)\mathtt{d}_Q(S_1, S_2).$$

If Q describes a continuous uniform distribution in $[0, L]^d$ (or sufficiently close to one, like points on a grid), then there exists an absolute constant $C > 0$ such that $C_q \leq C$ for all $q \in Q$, then in Lemma 5.5 $\sigma(q) \leq C_d\left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}}$ for all $q \in Q$, and in Theorem 5.5 $\mathfrak{S}_Q \leq C_d\left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}}$. So, for uniform distribution, the sample size of Q in Theorem 5.6 is independent from the size of Q, and for $d = 2$ the bound $\mathfrak{S}_Q = O(L/\rho)$ matches the lower bound in Lemma 5.4.

Corollary 5.1. *If Q describes the continuous uniform distribution over $[0, L]^d$, then the sample size in Theorem 5.6 can be reduced to $N = O\left(\left(\frac{L}{\rho}\right)^{\frac{2d}{2+d}}\frac{1}{\delta\varepsilon^2}\right)$.*

Remark 5.2. *To compute the upper bound of $\sigma(q)$ in Lemma 5.5, we need to compute C_q which can be obtained in $O(n \log n)$ time. For any fixed $q \in Q$, we sort $Q \setminus \{q\} = \{q_1, \cdots, q_{n-1}\}$ according*

to their l^∞ distance from q, so that $\|q - q_i\|_\infty \leq \|q - q_j\|_\infty$ for any $i < j$. Then for $i \in [n]$ we compute $\frac{r_i^d}{L^d}\frac{n}{i}$, where $r_i = \|q - q_i\|_\infty$ for $= i \in [n-1]$ and $r_n = L$, and choose the maximum value of $\frac{r_i^d}{L^d}\frac{n}{i}$ as C_q.

5.4 Strong Coresets for the Distance Between Trajectories

In this section, we study the distance $\mathtt{d}_Q$ defined on a subset of $\mathcal{S}(L)$: the collection of k-piecewise linear curves, and use the framework in [16] to construct a strong approximation for Q. We assume the multiset Q contains m distinct points $q_1, \cdots, q_m$, where each point q_i appears m_i times and $\sum_{i=1}^m m_i = n$. So, in this section Q will be viewed as a a set $\{q_1, \cdots, q_m\}$ (not a multiset) and each point $q \in Q$ has a weight $w(q_i) = \frac{m_i}{n}$.

Suppose $\mathcal{T}_k := \{\gamma = \langle c_0, \cdots, c_k\rangle \mid c_i \in \mathbb{R}^d\}$ is the collection of all piecewise-linear curves with k line segments in $\mathbb{R}^d$. For $\gamma = \langle c_0, \cdots, c_k\rangle \in \mathcal{T}_k$, $\langle c_0, \cdots, c_k\rangle$ is the sequence of $k+1$ critical points of γ. The value $\mathrm{dist}(q, \gamma) = \inf_{p\in\gamma} \|p - q\|$, and function $f_{\gamma_1,\gamma_2}(q) = (\mathrm{dist}(q, \gamma_1) - \mathrm{dist}(q, \gamma_2))^2$ are defined as before. We now use weights $w(q_i) = \frac{m_i}{n}$ $\left(\sum_{q\in Q} w(q) = 1\right)$ and the resulting distance is $\mathtt{d}_Q(\gamma_1, \gamma_2) = \left(\sum_{q\in Q} w(q) f_{\gamma_1,\gamma_2}(q)\right)^{\frac{1}{2}}$.

For $L > \rho > 0$, $Q = \{q_1, \cdots, q_m\} \subset \mathbb{R}^d$, we define

$$\mathcal{X}_k^d(L, \rho) := \{(\gamma_1, \gamma_2) \in \mathcal{T}_k \times \mathcal{T}_k \mid \gamma_1, \gamma_2 \in \mathcal{S}(L),\ \mathtt{d}_Q(\gamma_1, \gamma_2) \geq \rho\}.$$

We next consider the sensitivity adjusted weights $w'(q) = \frac{\sigma(q)}{\mathfrak{S}_Q} w(q)$ and cost function $g_{\gamma_1,\gamma_2}(q) = \frac{1}{\sigma(q)} \frac{f_{\gamma_1,\gamma_2}(q)}{\bar{f}_{\gamma_1,\gamma_2}}$. These use the general bounds for sensitivity in Lemma 5.5 and Theorem 5.5, with as usual $\bar{f}_{\gamma_1,\gamma_2} = \sum_{q\in Q} w(q) f_{\gamma_1,\gamma_2}(q)$. These induce an adjusted range space $(Q, \mathcal{T}'_{k,d})$ where each element is defined

$$T_{\gamma_1,\gamma_2,\eta} = \{q \in Q \mid w'(q) g_{\gamma_1,\gamma_2}(q) \leq \eta,\ \gamma_1, \gamma_2 \in \mathcal{X}_k^d(L, \rho)\}.$$

Now to apply the strong coreset construction of Braverman *et.al.* [16][Theorem 5.5] we only need to bound the shattering dimension of $(Q, \mathcal{T}'_{k,d})$.

Two recent results provide bounds on the VC-dimension of range spaces related to trajectories. Given a range space $(X, \mathcal{R})$ with VC-dimension ν and shattering dimension $\mathfrak{s}$, it is known that $\mathfrak{s} = O(\nu \log \nu)$ and $\nu = O(\mathfrak{s})$. So up to logarithmic factors these terms are bounded by each other. First Driemel *et.al.* [34] shows VC-dimension for a ground set of curves $\mathbb{X}_m$ of length m, with respect to metric balls around curves of length k, for various distance between curves. The most relevant case is where $m = 1$ (so the ground

set are points like Q), and the Hausdorff distance is considered, where the VC-dimension in $d = 2$ is bounded $O(k^2 \log(km)) = O(k^2 \log k)$ and is at least $\Omega(\max\{k, \log m\}) = \Omega(k)$. Second, Matheny *et.al.* [72] considered ground sets $\mathbb{X}_k$ of trajectories of length k, and ranges defined by geometric shapes which may intersect those trajectories anywhere to include them in a subset. The most relevant cases is when they consider disks, and show the VC-dimension is at most $O(d \log k)$, and have a proof that implies it is at least $\Omega(\log k)$; but this puts the complexity k on the ground set not the query. More specifically, neither of these cases directly imply the results for our intended range space, since ours involves a pair of trajectories.

Lemma 5.7. *The shattering dimension of range space $(Q, \mathcal{T}'_{k,d})$ is $O(k^3)$, for constant d.*

Proof. Suppose $(\gamma_1, \gamma_2) \in \mathcal{X}_k^d(L, \rho)$ and $\eta \geq 0$, where $\gamma_1 = \langle c_{1,0}, \cdots, c_{1,k} \rangle$ and $\gamma_2 = \langle c_{2,0}, \ldots, c_{2,k} \rangle$, then we can define the range $T_{\gamma_1,\gamma_2,\eta}$ as

$$\begin{aligned} T_{\gamma_1,\gamma_2,\eta} :=& \{q \in Q \mid w'(q) g_{\gamma_1,\gamma_2}(q) \leq \eta\} \\ =& \{q \in Q \mid w(q) f_{\gamma_1,\gamma_2}(q) \leq \mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2} \eta\} \\ =& \{q \in Q \mid w(q)(\mathrm{dist}(q, \gamma_1) - \mathrm{dist}(q, \gamma_2))^2 \leq \mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2} \eta\}. \end{aligned}$$

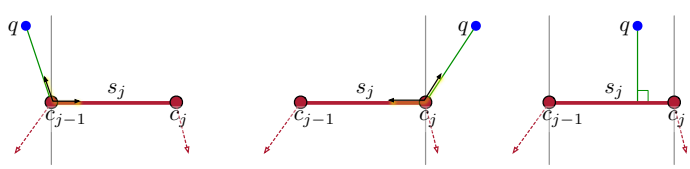

Figure 5.3: Illustration of the $\mathrm{dist}(q, s_j)$ from point q to segment s_j.

For a trajectory γ defined by critical points $c_0, c_1, \ldots, c_k$ for $j \in [k]$ define s_j as the segment between c_{j-1}, c_j and ℓ_j as the line extension of that segment. The distance between q and a segment s_j is illustrated in Figure 5.3 and defined

$$\xi_j := \mathrm{dist}(q, s_j) = \begin{cases} \mathrm{dist}(q, c_{j-1}), & \text{if } \langle c_j - c_{j-1},\ q - c_{j-1} \rangle \leq 0 \\ \mathrm{dist}(q, c_j), & \text{if } \langle c_{j-1} - c_j,\ q - c_j \rangle \leq 0 \\ \mathrm{dist}(q, \ell_j), & \text{otherwise} \end{cases}.$$

Then $\mathrm{dist}(q, \gamma) = \min_{j \in [k]} \xi_j$. For trajectories γ_1 and γ_2, specify these segment distances as $\xi_i^{(1)}$ and $\xi_i^{(2)}$, respectively. Then the expression for $T_{\gamma_1,\gamma_2,\eta}$ can be rewritten as

$$
\begin{aligned}
T_{\gamma_1,\gamma_2,\eta} &= \{q \in Q \mid w'(q) g_{\gamma_1,\gamma_2}(q) \le \eta\} \\
&= \{q \in Q \mid w(q)(\min_{j\in[k]} \xi_j^{(1)} - \min_{j\in[k]} \xi_j^{(2)})^2 \le \mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2}\eta\} \\
&= \cup_{j_1,j_2\in[k]} \{q \in Q \mid \xi_{j_1}^{(1)} \le \xi_j^{(1)}, \xi_{j_2}^{(2)} \le \xi_j^{(2)} \text{ for all } j \in [k],\ w(q)(\xi_{j_1}^{(1)} - \xi_{j_2}^{(2)})^2 \le \mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2}\eta\} \\
&= \bigcup_{j_1,j_2\in[k]} \begin{pmatrix} \left(\cap_{j\in[k],j\neq j_1} \{q \in Q \mid \xi_{j_1}^{(1)} \le \xi_j^{(1)}\}\right) \\ \cap\ \left(\cap_{j\in[k],j\neq j_2} \{q \in Q \mid \xi_{j_2}^{(2)} \le \xi_j^{(2)}\}\right) \\ \cap\ \{q \in Q \mid \sqrt{w(q)}(\xi_{j_1}^{(1)} - \xi_{j_2}^{(2)}) \le (\mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2}\eta)^{\frac{1}{2}}\} \\ \cap\ \{q \in Q \mid \sqrt{w(q)}(\xi_{j_2}^{(2)} - \xi_{j_1}^{(1)}) \le (\mathfrak{S}_Q \bar{f}_{\gamma_1,\gamma_2}\eta)^{\frac{1}{2}}\} \end{pmatrix}.
\end{aligned}
$$

This means set $T_{\gamma_1,\gamma_2,\eta}$ can be decomposed as the union and intersection of $O(k^3)$ simply-defined subsets of Q. Specifically looking at the last line, this can be seen as the union over $O(k^2)$ sets (the outer union), and the first two lines are the intersection of $O(k)$ sets, and the last two lines inside the union are the intersection with one set each.

Next we argue that each of these $O(k^3)$ simply defined subsets of Q can be characterized as an element of a range space. By standard combinatorics [11, 53] (and spelled out in Lemma 5.2), the bound of the shattering dimension of the entire range space is $O(k^3)$ times the shattering dimension of any of these simple ranges spaces.

To get this simple range space shattering dimension bound, we can use a similar linearization method as presented in the proof of Lemma 5.3. For any simple range space $\mathcal{R}$ determined by the set decomposition of $T_{\gamma_1,\gamma_2,\eta}$, we can introduce new variables $c_0 \in \mathbb{R}, z, c \in \mathbb{R}^{d'}$, where z depends only on q, and c_0, c_i depend only on γ_1, γ_2 and r, and d' only depends on d. Here, Q is a fixed set and thus $\mathfrak{S}_Q$ is a constant. By introducing new variables we can construct an injective map $\varphi : Q \mapsto \mathbb{R}^{d'}$, s.t. $\varphi(q) = z$. There is also an injective map from $\mathcal{R}$ to $\{\{z \in \varphi(Q) \mid c_0 + z^T c \le 0\} \mid c_0 \in \mathbb{R}, c \in \mathbb{R}^{d'}\}$. Since the shattering dimension of the range space $(\mathbb{R}^{d'}, \mathcal{H}^{d'})$, where $\mathcal{H}^{d'} = \{h \text{ is a halfspace in } \mathbb{R}^{d'}\}$, is $O(d')$, we have the shattering dimension of$(Q, \mathcal{R})$ is $O(d') \le C_d$ where C_d is a positive constant depending only on d. Piecing this all together we obtain $C_d k^3$ bound for the shattering dimension of $(Q, \mathcal{T}'_{k,d})$. □

Now, we can directly apply Lemma 5.7 and [16][Theorem 5.5] to get a $(\rho, \varepsilon, \delta)$-strong coreset for $\mathcal{X}_k^d(L, \rho)$.

Theorem 5.7. *Let $L > \rho > 0$, $Q \subset [0, L]^d$, and consider trajectory pairs $\mathcal{X}_k^d(L, \rho)$. Suppose $\sigma(q)$ and $\mathfrak{S}_Q$ are defined in Lemma 5.5 and Theorem 5.5 respectively. Then for $\delta, \varepsilon \in (0, 1)$ a σ-sensitive*

sampling of size $N = O(\frac{\mathfrak{S}_Q}{\varepsilon^2}(k^3 \log \mathfrak{S} + \log \frac{1}{\delta}))$ *provides* $\tilde{Q}$*, a strong* $(\rho, \varepsilon, \delta)$*-coreset; that is with probability at least* $1 - \delta$*, for all pairs* $\gamma_1, \gamma_2 \in \mathcal{X}_k^d(L, \rho)$ *we have*

$$(1 - \varepsilon)\mathtt{d}_Q(\gamma_1, \gamma_2) \leq \mathtt{d}_{\tilde{Q},W}(\gamma_1, \gamma_2) \leq (1 + \varepsilon)\mathtt{d}_Q(\gamma_1, \gamma_2).$$

5.5 Trajectory Reconstruction

In Section 5.4, we use Q to convert a piecewise-linear curve γ to a vector $v_Q(\gamma)$ in $\mathbb{R}^{|Q|}$, and in this section we study how to recover γ from Q and $v_Q(\gamma)$, and we only consider γ in $\mathbb{R}^2$.

Suppose $\mathcal{T} := \{\gamma = \langle c_0, \cdots, c_k \rangle |\ c_i \in \mathbb{R}^2, k \geq 1\}$ is the set of all piecewise-linear curves in $\mathbb{R}^2$. Let $\mathcal{T}_\tau$, $\mathcal{T}_\tau(\Omega)$ and G_η be define in the same way as in Section 4.3. For completeness, we relist two restrictions for trajectories in $\mathcal{T}_\tau$:

(R1) Each angle $\angle_{[c_{i-1}, c_i, c_{i+1}]}$ about an internal critical point c_i is non-zero (i.e., in $(0, \pi)$).

(R2) Each critical point c_i is τ*-separated*, that is the ball $B(c_i, \tau) = \{x \in \mathbb{R}^2 \mid \|x - c_i\| \leq \tau\}$ only intersects the two adjacent segments s_{i-1} and s_i of γ, or one adjacent segment for end points (i.e., only the s_1 for c_0 and s_k for c_k).

Suppose $\eta \leq \frac{\tau}{32}$, $Q = G_\eta \cap \Omega = \{q_1, \cdots, q_n\}$, $\gamma \in \mathcal{T}_\tau(\Omega)$, $v_i = \min_{p \in \gamma} \|q_i - p\|$ and $v_Q(\gamma) = (v_1, \ldots, v_n)$. We define some notations that are used in this section for the implied circle $C_i := \{x \in \mathbb{R}^2 |\ \|x - q_i\| = v_i\}$, the closed disk $B_i := \{x \in \mathbb{R}^2 |\ \|x - q_i\| \leq v_i\}$, and the open disk $\dot{B}_i := \{x \in \mathbb{R}^2 |\ \|x - q_i\| < v_i\}$ around each q_i or radius v_i. When the radius is specified as r (with perhaps $r \neq v_i$), then we, as follows, denote the associated circle $C_{i,r}$, closed disk $B_{i,r}$, and open disk $\dot{B}_{i,r}$ around q_i.

For Q, $\gamma \in \mathcal{T}_\tau(\Omega)$ and $v_Q(\gamma)$ we relist three observations in the proof of Theorem 4.6:

(O1) In any ball with radius less than τ, there is at most one critical point of γ; by (R2).

(O2) If a point moves along γ, then it can only stop or change direction at critical points.

(O3) For any $q_i \in Q$, γ cannot go into $\dot{B}_i$. Moreover, C_i must contain at least one point of γ, and if this point is not a critical point, then γ must be tangent to C_i at this point.

The restriction (R2) only implies if there is a critical point of γ, then in its neighborhood γ has at most two line segments. However, if there is no critical point in a region, then the

shape of γ can be very complicated in this region, so we need to first identify the regions that contain a critical point.

The entire algorithm is overviewed in Algorithm 5.2. For each critical point $c \in \gamma$, there exists $q \in Q$ such that $\mathrm{dist}(q, c) < \eta$. So to recover γ, we first traverse $\{q_i \in Q \mid v_i < \eta\}$ and use ISCRITICAL(q_i) (Algorithm 5.3) to solve the decision problem of if there is a critical point in $B_{i,3\eta}$. Whenever there is a critical point in $B_{i,3\eta}$, we then use FINDCRITICAL(q_i) (Algorithm 5.4) to find it – collectively, this finds all critical points of γ. Finally, we use DETERMINEORDER (Algorithm 5.5) to determine the order of all critical points of γ, which recovers γ.

Algorithm 5.2 Recover $\gamma \in \mathcal{T}_\tau(L)$ from Q and $v_Q(\gamma)$

```
Initialize Q_η := {q_i ∈ Q | v_i < η}, close set Q_r := ∅, endpoints E = ∅ and critical points
A := ∅.
for each q_i ∈ Q_η do
  if q_i ∈ Q_r or ISCRITICAL(q_i)=FALSE then
    continue
  Let (c, S) := FINDCRITICAL(q_i).
  if |S| = 1 then
    E := E ∪ {(c, S)}.      // c is an endpoint of γ
  Let A := A ∪ {(c, S)} and Q_r := Q_r ∪ (Q_η ∩ B_{c,16η}).      // aggregate critical points
return γ := DETERMINEORDER(E, A)
```

Existence of critical points.

In Algorithm 5.3, we consider the common tangent line of C_i and C_j for all q_j in a neighborhood of q_i. If no common tangent line can go through $B_{i,3\eta}$ without going into the interior of any other circle centered in $B_{i,3\eta}$, then it implies there is a critical point of γ in $B_{i,3\eta}$.

Algorithm 5.3 ISCRITICAL(q_i): Determine the existence of critical point in $B_{i,3\eta}$

```
for each q_j ∈ Q_{i,3η} \ {q_i} do
  Let ℓ_{i,j} be a common tangent line of C_i and C_j.
  if ℓ_{i,j} does not intersect with Ḃ_k for all q_k ∈ Q_{i,3η} \ {q_i, q_j} then
    return FALSE
return TRUE      // there must be a critical point in B_{i,3η}
```

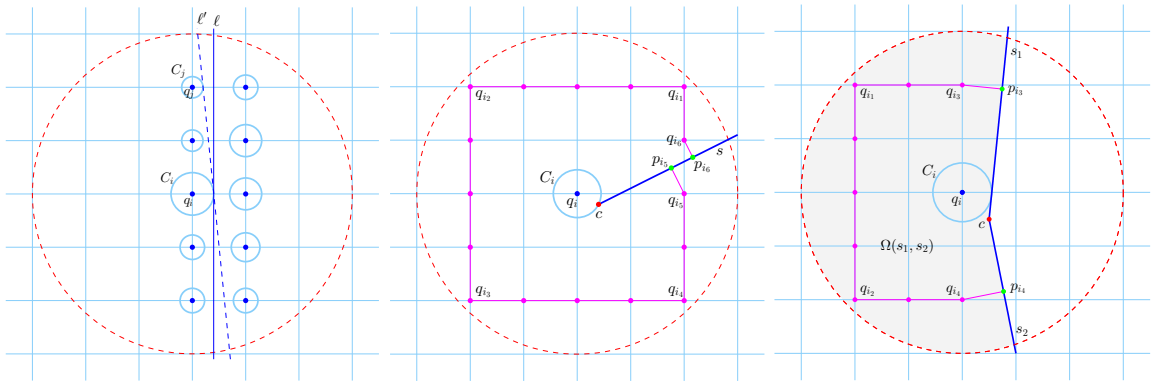

Figure 5.4: Left: l is tangent to C_i. Rotate l around C_i until it is tangent to some C_j. Center: c is an endpoint of γ. Right: c is an internal critical point of γ. In center and right figures, no tangent line of C_i can go through $B_{i,3\eta}$ without intersecting with the pink curve.

Lemma 5.8. *Suppose $q_i \in Q$ and $v_i < \eta$. If* ISCRITICAL(q_i) *(Algorithm 5.3) returns* TRUE*, then there must be a critical point of γ in $B_{i,3\eta}$. Moreover, for any critical point $c \in \gamma$ there exists some $q_i \in Q$ such that $v_i < \eta$ and* ISCRITICAL(q_i) *(Algorithm 5.3) returns* TRUE *for the input q_i.*

Proof. If Algorithm 5.3 returns TRUE, then no common tangent of C_i and C_j ($q_j \in Q_{i,3\eta}$) can go through $B_{i,3\eta}$ without intersecting with some $\dot{B}_k$ for $q_k \in Q_{i,3\eta}$. This implies no tangent line of C_i can go through $B_{i,3\eta}$ without intersecting with some $\dot{B}_k$ for $q_k \in Q_{i,3\eta}$. Otherwise, as shown in Figure 5.4(Left), suppose tangent line ℓ can go through $B_{i,3\eta}$, then we can rotate ℓ around C_i to line ℓ' s.t. ℓ' is tangent to some C_j ($q_j \in Q_{i,3\eta}$) but does not intersect with any $\dot{B}_k$ ($q_k \in Q_{i,3\eta}$), which leads to a contradiction. So, if there is no critical point on C_i then (O3) implies one line segment of γ must be tangent to C_i, but Algorithm 5.3 checks that no tangent line of C_i can go through $B_{i,3\eta}$ and thus from (O2) we know γ must have a critical point in $B_{i,3\eta}$.

If $c \in \gamma$ is a critical point, then there are two possibilities: c is an endpoint of γ, or c is an internal critical point of γ.

If c is an endpoint, let $q_i = (x_i, y_i)$ be the closest point in Q to c. Obviously we have $v_i < \eta$, and there is only one line segment s of γ in $B_{i,3\eta}$. We consider the points set $S_{i,2\eta} := \{(x_i + k_1\eta, y_i + k_2\eta) \mid \|(k_1, k_2)\|_\infty = 2\}$, i.e. the pink points in Figure 5.4(Center). Without loss of generality, we assume $q_{i_5} = (x_i + 2\eta, y_i)$ and $q_{i_6} = (x_i + 2\eta, y_i + \eta)$ are the two closest points in $S_{i,2\eta}$ to s, and their projection on s are p_{i_5} and p_{i_6} respectively (two green points in Figure 5.4(Center)). Let $q_{i_1} = (x_i + 2\eta, y_i + 2\eta)$, $q_{i_2} = (x_i - 2\eta, y_i + 2\eta)$,

$q_{i_3} = (x_i - 2\eta, y_i - 2\eta)$ and $q_{i_4} = (x_i + 2\eta, y_i - 2\eta)$ be the four pink corners. Since the radius of C_i is $v_i < \eta$, we know any tangent line of C_i must intersect with the piecewise-linear curve $\langle p_{i_6}, q_{i_6}, q_{i_1}, q_{i_2}, q_{i_3}, q_{i_4}, q_{i_5}, p_{i_5} \rangle$ before it passes completely through $B_{i,3\eta}$. However, the curve $\langle p_{i_6}, q_{i_6}, q_{i_1}, q_{i_2}, q_{i_3}, q_{i_4}, q_{i_5}, p_{i_5} \rangle$ is covered (except points p_{i_6} and p_{i_5}) by open disks $\dot{B}_k$ whose centers are in $q_k \in S_{i,2\eta} \subset Q_{i,3\eta}$. So, no tangent line of C_i can go through $B_{i,3\eta}$ without intersecting with some $\dot{B}_k$ for $q_k \in Q_{i,3\eta}$.

If c is an internal critical point, then there are two line segments s_1, s_2 in $B_{i,3\eta}$. From (R1) we know the angle between s_1 and s_2 is less than π, and we define $\Omega(s_1, s_2) := \{p \in B_{i,3\eta} \mid p \text{ is outside the interior angle region formed by } s_1 \text{ and } s_2\}$. Let $q_i = (x_i, y_i)$ be the closest point in $\Omega(s_1, s_2)$ to c, and $S_{i,2\eta}$ be defined in the same way as before. We have $v_i < \eta$. We consider the points set $S_{i,2\eta} \cap \Omega(s_1, s_2)$, i.e. those pink points in Figure 5.4(Right). Without loss of generality, we assume $q_{i_3} = (x_i, y_i + 2\eta)$ and $q_{i_4} = (x_i, y_i - 2\eta)$ are two closest points in $S_{i,2\eta} \cap \Omega(s_1, s_2)$ to s_1 and s_2 respectively, and their projection on s_1 and s_2 are p_{i_3} and p_{i_4} respectively (two green points in Figure 5.4(Right)). In this setting, let $q_{i_1} = (x_i - 2\eta, y_i + 2\eta)$ and $q_{i_2} = (x_i - 2\eta, y_i - 2\eta)$ be the corner points of $S_{i,2\eta}$. Since the radius of C_i is $v_i < \eta$ and the angle formed by s_1 and s_2 is less than π, we know any tangent line of C_i must intersect with the piecewise-linear curve $\langle p_{i_4}, q_{i_4}, q_{i_2}, q_{i_1}, q_{i_3}, p_{i_3} \rangle$ before go through $B_{i,3\eta}$. However, the curve $\langle p_{i_4}, q_{i_4}, q_{i_2}, q_{i_1}, q_{i_3}, p_{i_3} \rangle$ is covered by open disks $\dot{B}_k$ whose centers are $q_k \in S_{i,2\eta} \cap \Omega(s_1, s_2) \subset Q_{i,3\eta}$. So, we know no tangent line of C_i can pass entirely through $B_{i,3\eta}$ without intersecting with some $\dot{B}_k$ for $q_k \in Q_{i,3\eta}$.

Thus, if c is a critical point of γ, Algorithm 5.3 will return TRUE for some $q_i \in Q$ with $v_i < \eta$. □

Finding a critical point.

If there is a critical point c in $B_{i,3\eta}$, then using (R2) we know in the neighborhood of c, γ has a particular pattern: it either has one line segment, or two line segments. We will need two straightforward subfunctions:

- FCT (*Find Common Tangents*) takes in three grid points q_i, q_j, q_k, and returns the all common tangent lines of C_j and C_k which do not intersect the interior of disks $\dot{B}_l$ of an disk associated with a point $q_l \in Q_{i,8\eta}$. This generates a feasible superset of possible nearby line segments which may be part of γ.

- MOS (*Merge-Overlapping-Segments*) takes a set of line segments, and returns a smaller set, merging overlapping segments. This combines the just generated potential line segments of γ.

Now in Algorithm 5.4, for each pair $q_j, q_k \in B_{i,8\eta}$, we first use FCS to find the common tangent line of C_j, C_k that could be segments of γ, and then use MOS to reduce this set down to a minimal set of possibilities S_m. By definition, there must be a critical point c, and thus can be at most 2 actual segments of γ within $B_{i,8\eta}$, so we can then refine S_m. We first check if c is an endpoint, in which case there must be only one valid segment. If not, then there must be 2, and we need to consider all pairs in S_m. This check can be done by verifying that *every* C_k for $q_k \in Q_{i,8\eta}$ is tangent to the associated ray $\mathsf{ray}(s)$ (for an endpoint) or for the associated rays $\mathsf{ray}(s)$ and $\mathsf{ray}(s')$ for their associated segment pairs (for an internal critical point).

Algorithm 5.4 FINDCRITICAL(q_i): Find a critical point in $B_{i,3\eta}$

Let $Q_{i,8\eta} := Q \cap B_{i,8\eta}$ and $S_\mathrm{t} := \emptyset$.
for each pair $q_j, q_k \in Q_{i,8\eta}$ **do**
  $S_\mathrm{t} := S_\mathrm{t} \cup \mathrm{FCT}(q_i, q_j, q_k)$
$S_\mathrm{m} := \mathrm{MOS}(S_\mathrm{t})$.
for each $s \in S_\mathrm{m}$ **do**
  Extend s to ray $\mathsf{ray}(s)$ with endpoint c where it first enters $\dot{B}_k$ for some $q_k \in Q_{i,8\eta}$.
  if for all $q_j \in Q_{i,8\eta}$ either $c \in C_j$ or C_j is tangent to $\mathsf{ray}(s)$ (ENDPOINT) **then**
    return $(c, \{s\})$ // c is an endpoint of γ
for each pair $s, s' \in S_\mathrm{m}$ **do**
  Extend to lines $\ell(s), \ell(s')$.
  if $\ell(s)$ and $\ell(s')$ do not intersect in $B_{i,8\eta}$ **continue**
  Set $c = \ell(s) \cap \ell(s')$, and define rays from c containing s and s' as $\mathsf{ray}(s)$ and $\mathsf{ray}(s')$.
  if for all $q_k \in Q_{i,8\eta}$ either $c \in C_k$ or C_k is tangent to $\mathsf{ray}(s)$ or $\mathsf{ray}(s')$ (INTERNALPOINT) **then**
    return $(c, \{s, s'\})$ // c is an internal critical point of γ

Lemma 5.9. *Suppose $c' \in B_{i,3\eta}$ is a critical point of γ, and (c, S) is the output of* FINDCRITICAL(q_i) *(Algorithm 5.4), then $c = c'$. Moreover, $|S| = 1$ if and only if c is an endpoint of γ.*

Proof. Since $c' \in B_{i,3\eta}$ and $\eta < \frac{\tau}{32}$, we have $B_{i,8\eta} \subset B(c', \frac{\tau}{2})$. So, from (R2) we know in $B_{i,8\eta}$, γ either has one line segment which means c' is an endpoint, or has two line segments which means c' is an internal critical point.

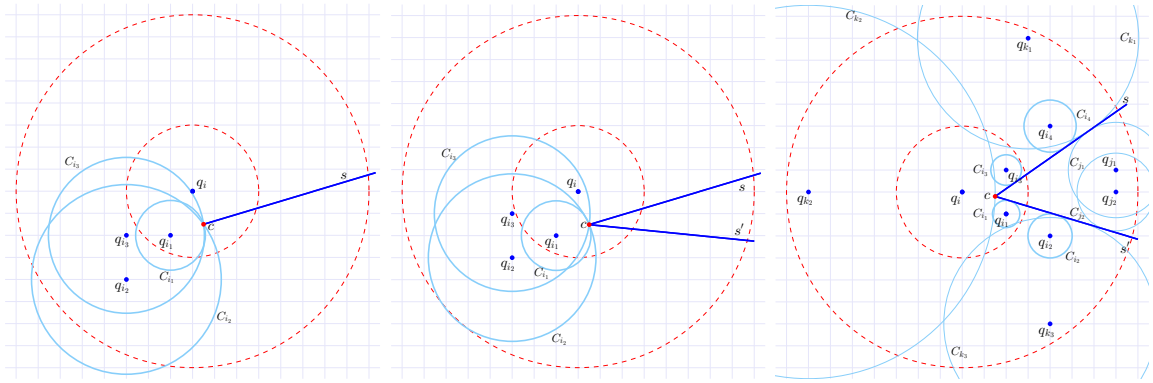

Figure 5.5: Left: $\{c\} = C_{i_1} \cap C_{i_2} \cap C_{i_3}$ and $B_{i_1} \subset B_{i_2} \cup B_{i_3}$. Center: the angle between s and s' is at most $\frac{\pi}{4}$ and $\{c\} = C_{i_1} \cap C_{i_2} \cap C_{i_3}$ and $B_{i_1} \subset B_{i_2} \cup B_{i_3}$. Right: C_{i_1}, C_{i_2} are tangent to s, and C_{i_3}, C_{i_4} are tangent to s', For each one of these four circles, any tangent line segment, except s, s', cannot be extended outside $B_{i,8\eta}$ without intersecting with any other circle.

If c' is an endpoint, then the line segment of γ must satisfy Condition ENDPOINT in Algorithm 5.4. Moreover, if in Algorithm 5.4 s satisfies Condition ENDPOINT, then c must be a critical point of γ. This is because, as show in Figure 5.5(Left), there exists three points $q_{i_1}, q_{i_2}, q_{i_3} \in Q_{i,8\eta}$ such that $\{c\} = C_{i_1} \cap C_{i_2} \cap C_{i_3}$ and $B_{i_1} \subset B_{i_2} \cup B_{i_3}$ and the tangent of C_{i_1} at c intersects with $\dot{B}_{i_2} \cup \dot{B}_{i_3}$. This can be seen by observing there must exists points $q_{i_2}, q_{i_3} \in Q_{i,8\eta}$ which are (i) on the opposite side from s of the perpendicular to s through c, (ii) are a distance at least 3η from c, and (iii) within a distance of 3η from each other. This implies there exists another point $q_{i_1} \in Q \cap B_{i_2} \cap B_{i_3}$ and with $v_{i_1} \leq 2\eta$. Hence B_{i_1} must be contained in $B_{i_2} \cup B_{i_3}$. Thus, (O3) implies c is a critical point of γ, and from (O1) we know $c = c'$.

If c' is an interior point, then as show in Figure 5.5(Center and Right), no line segment can satisfy Condition ENDPOINT in Algorithm 5.4, so the algorithm will not stop before the third loop. Then the two line segments of γ with c' as the common endpoint can satisfy Condition INTERNALPOINT. Moreover, if s and s' satisfy Condition INTERNALPOINT, then we will show c must be a critical point of γ. There are two possibilities: the angle between s and s at most $\frac{\pi}{4}$, or greater than $\frac{\pi}{4}$.

If the angle is less than or equal to $\frac{\pi}{4}$, then as shown in Figure 5.5(Center), there exists three points $q_{i_1}, q_{i_2}, q_{i_3} \in Q_{i,8\eta}$ such that $\{c\} = C_{i_1} \cap C_{i_2} \cap C_{i_3}$ and $B_{i_1} \subset B_{i_2} \cup B_{i_3}$ and the tangent of C_{i_1} at c intersects with $\dot{B}_{i_2} \cup \dot{B}_{i_3}$. This follows by the same argument as when c is an endpoint. So, (O3) implies c is a critical point of γ, and from (O1) we know $c = c'$.

If the angle is greater than $\frac{\pi}{4}$, then as shown in Figure 5.5(Right), there exists four points $q_{i_1}, q_{i_2}, q_{i_3}, q_{i_4} \in Q_{i,8\eta}$ outside the interior angular region, and such that C_{i_1}, C_{i_2} are tangent to s', and C_{i_3}, C_{i_4} are tangent to s. Moreover, these four circles can be chosen to not intersect with each other. Next we can argue that because the angle is sufficiently large, we can block a path from c' to outside of $B_{i,8\eta}$ both inside the interior angular region, and outside it. Outside this region, we can choose three points in $q_{k_1}, q_{k_2}, q_{k_3} \in Q_{i,8\eta}$ of which C_{k_1} is incident to $\mathsf{ray}(s)$, C_{k_2} is incident to c', and C_{k_3} is incident to $\mathsf{ray}(s')$; and that $\dot{B}_{k_1}$ and $\dot{B}_{k_2}$ intersect and $\dot{B}_{k_2}$ and $\dot{B}_{k_3}$ intersect. Similarly, inside the interior angular region, we can chose two points $q_{j_1}, q_{j_2} \in Q_{i,8\eta}$ so C_{j_1} and C_{j_2} are incident to $\mathsf{ray}(s)$ and $\mathsf{ray}(s')$, respectively, and that $\dot{B}_{j_1}$ and $\dot{B}_{j_2}$ intersect. These two sets of points blocks any other straight path from c' (required by (O2)) from existing $B_{i,8\eta}$ (required by (O1)) without entering the interior of some $\dot{B}_k$. And the first four points $q_{i_1}, q_{i_2}, q_{i_3}, q_{i_4}$ ensures that this c' is unique (by (O1)) and $c' = c$ must be a critical point on γ. □

Using Algorithm 5.3 and 5.4 we can find all critical points (E, A) with associated line segments of γ, so the final step is to use function DETERMINEORDER(E, A) (Algorithm 5.5) to determine their order, as we argue it will completely recover γ.

Algorithm 5.5 DETERMINEORDER(E, A): Determine the order of critical points

Choose any $(c_0, S_0) \in E$, let $k = |A| - 1$, $A := A \setminus \{(c_0, S_0)\}$, $s_1 \in S_0$ and $\gamma := \langle c_0 \rangle$.
for $i = 1$ **to** k **do**
  Find closest c from $(c, S) \in A$ to c_{i-1} such that c is on $\mathsf{ray}(s_i)$, and let $A := A \setminus \{(c, S)\}$.
  Append $c_i = c$ to γ, and if $i < k$ then let $s_{i+1} = s$ where $s \in S$ is not parallel with s_i.
return γ

Theorem 5.8. *Suppose $Q = G_\eta \cap \Omega$, $\eta \leq \frac{\tau}{32}$, and $v_Q(\gamma)$ is generated by Q and $\gamma \in \mathcal{T}_\tau(\Omega)$, then Algorithm 5.2 can recover γ from $v_Q(\gamma)$ in $O(|Q| + k^2)$ time, where k is the number of line segments of γ.*

Proof. From Lemmas 5.3 and 5.4 we know Algorithms 5.3 and 5.4 identify all critical points of γ, and the line segments of γ associated with each critical point. So we only need to show Algorithm 5.5 determines the correct order of critical points. This is because if a point moves along γ it cannot stop or change direction until it hits a critical point (Observation (O2)),

and when it hits a critical point it has to stop or change direction, otherwise it will violate (R1) or (R2). So, Algorithm 5.5 starts from an endpoint and moves along the direction of line segment associated with it, and changes the direction only after arriving at the next critical point, until all critical points are visited. This gives the correct order of critical points of γ.

Moreover, the running time of Algorithm 5.3 and 5.4 are constant, since they both only examine a constant number of points, circles, etc in each $B_{i,3\eta}$ or $B_{i,8\eta}$. And these can be retrieved using the implicit grid structure in constant time. Thus the **for** loop in Algorithm 5.2 takes $O(|Q|)$ time. The final Algorithm 5.5 to recover the order takes $O(k^2)$ time, since a constant fraction of steps need to check a constant fraction of all critical points in A. So, the total running time of this algorithm is $O(|Q| + k^2)$. $\square$

CHAPTER 6

CONCLUSION

In this dissertation, we mainly study how to analyze and summarize the uncertain data points, how to sketch lines, trajectories and other geometric shapes, and the application of this sketched representation. We analyze the uncertain data by studying the robust estimators, especially the median, on the data set. We design an efficient deterministic algorithm to construct ε-approximate coreset for Tukey median and geometric median on a set of uncertain data points in high dimensional space. Moreover, for robust estimators associated with bounded VC-dimension range spaces in a general metric space, we design a random algorithm to approximate them on uncertain data. We also argue that although we can use such distributions to calculate a single-point representation of these distributions, it is not very stable to the input distributions, and serves as a poor representation when the true scenario is multi-modal; hence further motivating our distributional approach.

Moreover, for robust estimators, we give a formal definition for break down point and study the robustness of composite estimators. We show the composition of two or more estimators is usually less robust than each individual estimator, and give the condition under which the breakdown point of the composite estimator is the product of the breakdown points of the individual estimators. This result can be applied in understanding complicated data analysis pipelines and provide worst case guarantees.

Another contribution of this work is a vectorized representations based on landmarks for geometric objects. Using this representation, we introduce a new family of landmark-based distances $\mathtt{d}_Q$ for lines, hyperplanes and general shapes. These distances have nice mathematical properties, are easily to compute, and can be applied in trajectory clustering and classification, where they demonstrate a strong competitiveness and advantages against other distances. Moreover, when the landmark set Q is very large, we can use sensitivity sampling method to sample a small subset from Q to approximate $\mathtt{d}_Q$ for pairs

of general geometric objects, and for hyperplanes and trajectories we can construct a strong approximation of Q and bound the sample size. For trajectories from a mildly restricted family, we design an algorithm to exactly recover them from landmarks and their vectorized representations. We believe more interesting properties and applications of this vectorized representation and landmark-based distance are worthy of study.

APPENDIX A

THE APPENDIX OF CHAPTER 2

A.1 The Size of $\hat{T}$ Based on $\hat{\mathsf{cost}}$

For a given positive number ε and a set of uncertain points $\mathcal{P} = \{P_1, \cdots, P_n\}$ where $P_i = \{p_{i,1}, \cdots, p_{i,k}\} \subset \mathbb{R}$, $i \in [n]$, if we define $\hat{\mathsf{cost}}(x) = \frac{1}{n}\sum_{i=1}^{n} \min_{1\le j\le k} |x - p_{i,j}|$ and try to find a set $\hat{T}$ such that for any $Q \Subset \mathcal{P}$, there exists $x \in \hat{T}$ s.t. $|x - m_Q| \le \varepsilon\hat{\mathsf{cost}}(m_Q)$, then for some fixed $\varepsilon > 0$, the size of $\hat{T}$ may satisfy $|\hat{T}| = \Omega(nk)$.

In fact, for this data set: $\varepsilon = \frac{1}{4}$, $k = 2$, $p_{i,1} = 1 - \frac{1}{2^{i-1}}$ and $p_{i,2} = 1$ for all $i \in [n]$, we have

$$\begin{aligned}
\hat{\mathsf{cost}}(p_{i,1}) =& \frac{1}{n}\left(\sum_{j=1}^{i-1}(p_{j,2} - p_{i,1}) + \sum_{j=i+1}^{n}(p_{j,1} - p_{i,1})\right) \\
=& \frac{1}{n}\left(\sum_{j=1}^{i-1}\left(1 - (1 - \frac{1}{2^{i-1}})\right) + \sum_{j=i+1}^{n}\left(1 - \frac{1}{2^{j-1}} - (1 - \frac{1}{2^{i-1}})\right)\right) \\
=& \frac{1}{2^{i-1}} + \frac{1}{n}\left(\frac{1}{2^{n-1}} - 2\frac{1}{2^{i-1}}\right) < \frac{1}{2^{i-1}},
\end{aligned}$$

which implies

$$\varepsilon\hat{\mathsf{cost}}(p_{i,1}) + \varepsilon\hat{\mathsf{cost}}(p_{i+1,1}) < \frac{1}{4}\frac{1}{2^{i-1}} + \frac{1}{4}\frac{1}{2^{i}} < \frac{1}{2^{i}} = p_{i+1,1} - p_{i,1}.$$

So we have $[p_{i,1} - \varepsilon\hat{\mathsf{cost}}(p_{i,1}), p_{i,1} + \varepsilon\hat{\mathsf{cost}}(p_{i,1})] \cap [p_{i+1,1} - \varepsilon\hat{\mathsf{cost}}(p_{i+1,1}), p_{i+1,1} + \varepsilon\hat{\mathsf{cost}}(p_{i+1,1})] = \emptyset$ for $i \in [n]$, which implies $|\hat{T}| \ge n$.

Now, if we consider $n = 1, 2, 3, \cdots$, $k = 2, 4, 6, \cdots$ and $p_{i,j} = \frac{1}{2}(3j - 1) - \frac{1}{2^{i-1}}$, $p_{i,j+1} = \frac{1}{2}(3j - 1)$ for $j = 1, 3, 5, \cdots k - 1$ and $i \in [n]$, then is easy to check $|\hat{T}| \ge \frac{1}{2}kn$. Therefore, we have $|\hat{T}| = \Omega(nk)$.

A.2 A Property of Geometric Median

To prove the result of Lemma 2.1, we need the following property of geometric median. Although this result is stated on Wikipedia, we have not found a proof in the literature, so we present it here for completeness.

Lemma A.1. *Suppose p is the geometric median of $Q = \{q_1, \cdots, q_n\} \subset \mathbb{R}^d$, and $(x_1, \cdots, x_d)$ and $(x_{i,1}, \cdots, x_{i,d})$ are the coordinates of p and q_i respectively, then we have $|\sum_{q_i \in Q \setminus \{p\}} \frac{x_j - x_{i,j}}{\|q-p\|}| \leq |Q \cap \{p\}|$ for any $j \in [d]$.*

Proof. We introduce the notation $f(y) = f_1(y) + f_2(y)$ where $f_1(y) = \sum_{q \in Q \setminus \{p\}} \|q - y\|$ and $f_2(y) = \sum_{q \in Q \cap \{p\}} \|q - y\|$. Suppose $v_j \in \mathbb{R}^d$ is a vector such that its j-th component is one and all other components are zero. Since p is the global minimum point of f, for any $j \in [d]$ there exists $\delta_j > 0$ such that

$$f(p + \varepsilon v_j) \geq f(p) \text{ and } f(p - \varepsilon v_j) \geq f(p), \quad \forall\, \varepsilon \in [0, \delta_j),$$

which implies

$$f_1(p + \varepsilon v_j) + f_2(p + \varepsilon v_j) \geq f_1(p) + f_2(p), \ \ \forall\, \varepsilon \in [0, \delta_j), \tag{A.1}$$

and

$$f_1(p - \varepsilon v_j) + f_2(p - \varepsilon v_j) \geq f_1(p) + f_2(p), \ \ \forall\, \varepsilon \in [0, \delta_j). \tag{A.2}$$

Since $f_2(p) = 0$, from (A.1) we have $\frac{1}{\varepsilon}(f_1(p + \varepsilon v_j) - f_1(p)) \geq -\frac{1}{\varepsilon} f_2(p + \varepsilon v_j) = -|Q \cap \{p\}|$. Letting $\varepsilon \to 0+$, we obtain $\frac{\partial f_1(p)}{\partial x_j} \geq -|Q \cap \{p\}|$ which implies

$$\sum_{q_i \in Q \setminus \{p\}} \frac{x_j - x_{i,j}}{\|q - p\|} \geq -|Q \cap \{p\}|. \tag{A.3}$$

Similarly, using (A.2) we can obtain

$$\sum_{q_i \in Q \setminus \{p\}} \frac{x_j - x_{i,j}}{\|q - p\|} \leq |Q \cap \{p\}|. \tag{A.4}$$

Thus, conclusion of this lemma is obtained from (A.3) and (A.4). □

The bound in Lemma A.1 is tight. For example, we consider $Q = \{(-2, 0), (-1, 0), (0, 0), (1, 0), (-1, 1), (-1, -1)\} \subset \mathbb{R}^2$, then $p = (-1, 0)$ is the geometric median of Q and $|\sum_{q=(x_q, y_q) \in Q \setminus \{p\}} \frac{-1 - x_q}{\|q-p\|}| = 1 = |Q \cap p|$.

APPENDIX B

THE APPENDIX OF CHAPTER 4

B.1 Metric Properties for $\mathtt{d}_Q$ on Trajectories

In this section, we prove Theorem 4.6 for $\mathtt{d}_Q$. We first introduce the following lemma.

Lemma B.1. *As shown in Figure B.1, suppose the line ℓ_2 passes through q_1 and c, ℓ_1 is perpendicular to ℓ_2 at q_1, and c is on the right side of ℓ_1. If q_2 is outside the circle $C(q_1, \|q_1 - c\|)$, on the left side of ℓ_2 and above ℓ_2 (the yellow-shaded region)), and q_3 is outside the circle $C(q_1, \|q_1 - c\|)$, on the left side of ℓ_2 and below ℓ_2 (the pink-shaded region), then we have $B(q_1, \|q_1 - c\|) \subset B(q_2, \|q_2 - c\|) \cup B(q_2, \|q_2 - c\|)$.*

Proof. We use q_1 as the origin, ℓ_2 as the x-axis and ℓ_1 as the y-axis to build a coordinate system, and assume the coordinates of c, q_2 and q_3 are $(r, 0)$, (x_2, y_2) and (x_3, y_3) respectively. So, we have $x_2^2 + y_2^2 > r^2$, $x_2^2 + y_2^2 > r^2$ and $x_2, x_3 < 0$, $y_2 > 0$ and $y_3 < 0$. Our goal is to prove if $x^2 + y^2 \leq r^2$ then either

$$(x - x_2)^2 + (y - y_2)^2 \leq (x_2 - r)^2 + y_2^2, \tag{B.1}$$

or

$$(x - x_3)^2 + (y - y_3)^2 \leq (x_3 - r)^2 + y_3^2. \tag{B.2}$$

If $y \geq 0$, then from $x \leq r, x_2 < 0, y_2 > 0$ we have $(r - x)x_2 \leq yy_2$, which is equivalent to $-2xx_2 - 2yy_2 \leq -2rx_2$. Since $x^2 + y^2 \leq r^2$, we obtain $x^2 - 2xx_2 + y^2 - 2yy_2 \leq -2rx_2 + r^2$, which implies (B.1) is true. Similarly, if $y \leq 0$ then we can show (B.2) is true. Thus, the proof is completed.

□

Now, we can give the proof of Theorem 4.6 for $\mathtt{d}_Q$.

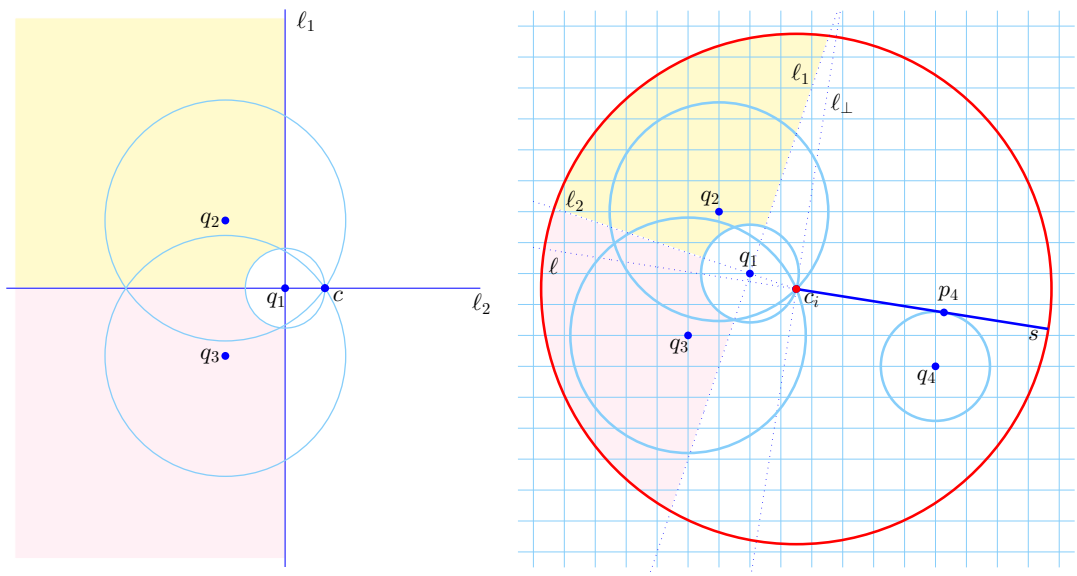

Figure B.1: Left: $\ell_1 \perp \ell_2$ and $B(q_1, \|q_1 - c\|) \subset B(q_2, \|q_2 - c\|) \cup B(q_3, \|q_3 - c\|)$. Right: c_i is a critical point of $\gamma^{(1)}$ and $B(q_1, \|q_1 - c_i\|) \subset B(q_2, \|q_2 - c_i\|) \cup B(q_3, \|q_3 - c_i\|)$.

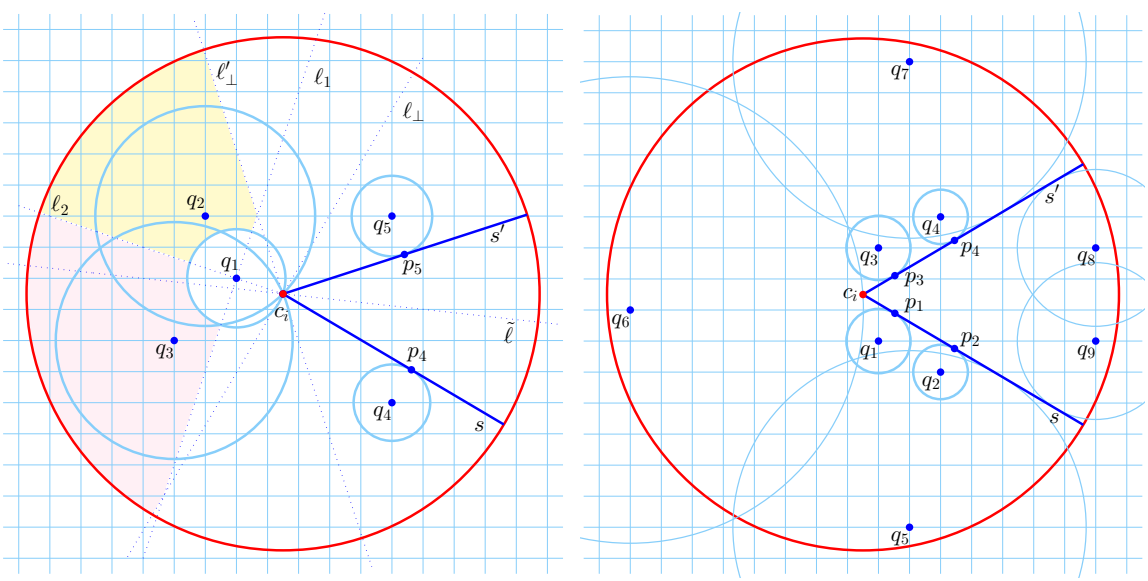

Figure B.2: Left: c_i is a critical point of $\gamma^{(1)}$ and $B(q_1, \|q_1 - c_i\|) \subset B(q_2, \|q_2 - c_i\|) \cup B(q_3, \|q_3 - c_i\|)$. Right: $B(q_1, \|q_1 - p_1\|)$, $B(q_2, \|q_2 - p_2\|)$ are tangent to s, and $B(q_3, \|q_3 - p_3\|)$, $B(q_4, \|q_4 - p_4\|)$ are tangent to s'. For each one of these four circles, any tangent line segment, except s, s' cannot be extended outside $B(c_i, \frac{\tau}{2})$ without intersecting with any other circle.

Proof. Suppose $\mathtt{d}_Q(\gamma^{(1)}, \gamma^{(2)}) = 0$, we only need to prove $\gamma^{(1)} = \gamma^{(2)}$. We draw a ball $B(c_i, \frac{1}{2}\tau)$ at a critical point c_i $(1 \le i \le k-1)$ of $\gamma^{(1)}$. There are three possibilities.

Case 1. As shown in Figure B.1(Right), c_i is an endpoint of $\gamma^{(1)}$, and $B(c_i, \frac{\tau}{2})$ contains

one line segment s of $\gamma^{(1)}$. In this case, we assume s is part of line ℓ, and draw a line $\ell_\perp$ through c_i which is perpendicular to ℓ. Then, we choose a point q_1 from $Q \cap B(c_i, \frac{\tau}{2})$, which is on the left side of $\ell_\perp$, close to ℓ and satisfies $\|q_1 - c\| < 2\eta$. Suppose ℓ_2 is the line through q_1 and c_i, and ℓ_1 is perpendicular to ℓ_2 at q_1. We choose a point $q_2 \in Q \cap B(c_i, \frac{\tau}{2})$ from the region that is outside $B(q_1, \|q_1 - c_i\|)$, on the left side of ℓ_1 and $\ell_\perp$, and above ℓ_2 (the yellow-shaded region), and choose a point $q_3 \in Q \cap B(c_i, \frac{\tau}{2})$ from the region that is outside $B(q_1, \|q_1 - c_i\|)$, on the left side of ℓ_1 and $\ell_\perp$, and below ℓ_2 (the pink-shaded region). Obviously, $\{c_i\} = C(q_1, \|q_1 - c_i\|) \cap C(q_2, \|q_2 - c_i\|) \cap C(q_3, \|q_3 - c_i\|)$, and from Lemma B.1, we know $B(q_1, \|q_1 - c_i\|) \subset B(q_2, \|q_2 - c_i\|) \cup B(q_3, \|q_3 - c_i\|)$. So, c_i must be on $\gamma^{(2)}$. Since the tangent line of $C(q_1, \|q_1 - c_i\|)$ at c_i goes into the interior of $B(q_2, \|q_2 - c_i\|)$ and $B(q_3, \|q_3 - c_i\|)$, from $(O3)$ we know c_i must be a critical point of $\gamma^{(2)}$. There also exists $q_4 \in B(c_i, \frac{\tau}{2})$ and $p_4 \in s$ such that $B(q_4, \|q_4 - p_4\|)$ is tangent to s at point p_4. From (O3) and (O1) we know the tangent line segment of $C(q_4, \|q_4 - p_4\|)$ through c_i must be a part of $\gamma^{(2)}$, and this tangent line segment must be s because the other tangent line segment through c_i intersects with other circles. Thus, s is a part of $\gamma^{(2)}$.

Case 2. As shown in Figure B.2(Left), c_i is an internal of $\gamma^{(1)}$, $B(c_i, \frac{\tau}{2})$ contains two line segments s, s' of $\gamma^{(1)}$, and the angle between s, s' is at most $\frac{\pi}{4}$. In this case, we assume $\tilde{\ell}$ is the line bisecting the angle formed by s and s', and draw two lines $\ell_\perp$ and $\ell'_\perp$ which is perpendicular to s and s' at c_i respectively. Then, we choose a point q_1 from $Q \cap B(c_i, \frac{\tau}{2})$, which is on the left side of $\ell_\perp$ and $\ell'_\perp$, close to $\tilde{\ell}$ and satisfies $\|q_1 - c\| < 2\eta$. Suppose ℓ_2 is the line through q_1 and c_i, and ℓ_1 is perpendicular to ℓ_2 at q_1. We choose a point $q_2 \in Q \cap B(c_i, \frac{\tau}{2})$ from the region that is outside $B(q_1, \|q_1 - c_i\|)$, on the left side of ℓ_1, $\ell_\perp$ and $\ell'_\perp$ and above ℓ_2 (the yellow-shaded region), and choose a point $q_3 \in Q \cap B(c_i, \frac{\tau}{2})$ from the region that is outside $B(q_1, \|q_1 - c_i\|)$, on the left side of ℓ_1, $\ell_\perp$ and $\ell'_\perp$ and below ℓ_2 (the pink-shaded region). Obviously, $\{c_i\} = C(q_1, \|q_1 - c_i\|) \cap C(q_2, \|q_2 - c_i\|) \cap C(q_3, \|q_3 - c_i\|)$, and from Lemma B.1, we know $B(q_1, \|q_1 - c_i\|) \subset B(q_2, \|q_2 - c_i\|) \cup B(q_3, \|q_3 - c_i\|)$. So, c_i must be on $\gamma^{(2)}$. Since the tangent line of $C(q_1, \|q_1 - c_i\|)$ at c_i goes into the interior of $B(q_2, \|q_2 - c_i\|)$ and $B(q_3, \|q_3 - c_i\|)$, from $(O3)$ we know c_i must be a critical point of $\gamma^{(2)}$. There also exists $q_4, q_5 \in B(c_i, \frac{\tau}{2})$ and $p_4 \in s$, $p_5 \in s'$ such that $B(q_4, \|q_4 - p_4\|)$ is tangent to s at point p_4, and $B(q_5, \|q_5 - p_5\|)$ is tangent to s' at point p_5. Using the similar argument in Case 1,we can show s and s' both belong to $\gamma^{(2)}$.

Case 3. As shown in Figure B.2(Right), c_i is an internal of $\gamma^{(1)}$, $B(c_i, \frac{\tau}{2})$ contains two line segments s, s' of $\gamma^{(1)}$, and the angle between s, s' is greater than $\frac{\pi}{4}$. In this case, we choose four points q_1, q_2, q_3, q_4 from $Q \cap B(c_i, \frac{\tau}{2})$ such that the circles with center q_1, q_2 are tangent to s at p_1, p_2, and the circles with center q_3, q_4 are tangent to s' at p_3, p_4. Moreover, we can require $\|q_j - c_j\| \leq \eta$ for $1 \leq j \leq 4$ and these four circles do not intersect with each other. Then, we can choose three points q_5, q_6, q_7 outside the angle region formed by s and s', and two points q_8, q_9 inside this angle region. Using $C_{j'}$ ($5 \leq j' \leq 9$) to represent the circles corresponding to these five points, we can choose these points close to the boundary of $B(c_i, \frac{\tau}{2})$, and require C_6 contains c_i, C_5, C_9 are tangent to s, C_7, C_8 are tangent to s', and $C_5 \cap C_6 \neq \emptyset$, $C_6 \cap C_7 \neq \emptyset$, and $C_8 \cap C_9 \neq \emptyset$. Thus, any tangent line segment of $C(q_j, \|q_j - p_j\|)$ ($1 \leq j \leq 4$), except s, s', can not be extended outside $B(c_i, \frac{\tau}{2})$ without intersecting with $\cup_{5 \leq j' \leq 9} C_{j'}$. From (O3) and (O1) we know $\gamma^{(2)}$ must be tangent to $C(q_1, \|q_1 - p_1\|)$ or $C(q_2, \|q_2 - p_2\|)$, and without loss of generality we assume a tangent line segment of $C(q_1, \|q_1 - p_1\|)$ is a part of $\gamma^{(2)}$. Since (O1), (O2) imply this tangent line segment must be extended outside $B(c_i, \frac{\tau}{2})$ without going into the interior of any other circle, we know $s \cap B(q_1, \delta)$ is a part of $\gamma^{(2)}$ for some $\delta > 0$. Similarly, we have $s \cap B(q_3, \delta)$ is a part of $\gamma^{(2)}$ for some $\delta > 0$. Since there is at most one critical point of $\gamma^{(2)}$ in $B(c_i, \frac{\tau}{2})$, from (O2) we know c_i must be a critical point of $\gamma^{(2)}$. Thus, s and s' both belong to $\gamma^{(2)}$.

From the discussion of above three cases, we know $\gamma^{(2)}$ overlaps with $\gamma^{(1)}$ in the ball $B(c_i, \frac{\tau}{2})$, and a similar argument leads to $\gamma^{(1)} = \gamma^{(2)}$.

□

B.2 Common Distance Measurements for Trajectories

In this section, we briefly introduce the definition of Euclidian distance, discrete Frechet distance and dynamic time warping distance. Suppose $\gamma^{(1)}$ and $\gamma^{(2)}$ are two trajectories in $\mathbb{R}^2$ with critical points $c_0^{(1)}, c_1^{(1)}, ...c_{k_1}^{(1)}$ and $c_0^{(2)}, c_1^{(2)}, ...c_{k_2}^{(2)}$ respectively.

Euclidean Distance. It requires $k_1 = k_2$ and takes the average Euclidean distance between corresponding critical points.

$$\mathrm{Eu}(\gamma^{(1)}, \gamma^{(2)}) = \frac{1}{k_1} \sum_{i=0}^{k_1} \|c_i^{(1)} - c_i^{(2)}\|.$$

Discrete Frechet Distance. It measures the similarity between two piecewise-linear curves

by taking into account their location and time ordering. Here, we introduce its definition in [38]. Suppose $\mathcal{A} = \{a_0, a_1, \cdots, a_m\} \subset \{0, 1, \cdots, k_1\}$, $\mathcal{B} = \{b_0, b_1, \ldots, b_m\} \subset \{0, 1, \cdots, k_2\}$, and $a_0 = b_0 = 0$, $a_m = k_1$, $b_m = k_2$. If for each $i \in \{0, \cdots, k_1 - 1\}$ we have $a_{i+1} = a_i$ or $a_{i+1} = a_i + 1$, and for each $i \in \{0, \cdots, k_2 - 1\}$, we have $b_{i+1} = b_i$ or $b_{i+1} = b_i + 1$, then we say $\mathcal{A}$ and $\mathcal{B}$ can determine a coupling $\mathcal{L}$ between $\gamma^{(1)}$ and $\gamma^{(2)}$, which is a sequence $(c^{(1)}_{a_0}, c^{(2)}_{b_0}), (c^{(1)}_{a_1}, c^{(2)}_{b_1}), \cdots, (c^{(1)}_{a_m}, c^{(2)}_{b_m})$. We define the *length* of $\mathcal{L}$ as $\|\mathcal{L}\| = \max_{0 \le i \le m} \|c^{(1)}_{a_i} - c^{(2)}_{b_i}\|$. The discrete Frechet distance is defined as:

$$\mathrm{dF}(\gamma^{(1)}, \gamma^{(2)}) = \min\{\|\mathcal{L}\| \mid \mathcal{L} \text{ is a a coupling between } \gamma^{(1)} \text{ and } \gamma^{(2)}\}.$$

Dynamic Time Warping (DTW) Distance. DTW [93] is an algorithm to find the optimal matching between the critical points of two trajectories, and it does not require $k_1 = k_2$. It is defined and computed by the recursion formula: $D(i, j) = \|c^{(1)}_i - c^{(2)}_j\| + \min\big(D(i-1, j),\ D(i-1, j-1),\ D(i, j-1)\big)$, where $D(0, j) = \|c^{(1)}_0 - c^{(2)}_j\|$, $D(i, 0) = \|c^{(1)}_i - c^{(2)}_0\|$, and DTW distance between $\gamma^{(1)}$ and $\gamma^{(2)}$ is defined as $\mathrm{DTW}(\gamma^{(1)}, \gamma^{(2)}) = D(k_1, k_2)$.

Discrete Hausdorff Distance. It measure the spatial similarity between two trajectories [71]:

$$\mathrm{dH}(\gamma^{(1)}, \gamma^{(2)}) = \max(d(\gamma^{(1)}, \gamma^{(2)}), d(\gamma^{(2)}, \gamma^{(1)}))$$

where $d(\gamma^{(1)}, \gamma^{(2)}) = \max_{0 \le i \le k_1} \min_{0 \le j \le k_2} \|c^{(1)}_i - c^{(2)}_j\|$.

Longest Common Subsequence Distance. It finds the alignment between two sequences that maximize the length of common subsequence. Let $\mathrm{Head}(\gamma^{(1)})$ be the first $k_1 - 1$ critical points of $\gamma^{(1)}$, and $\mathrm{Head}(\gamma^{(2)})$ be the first $k_2 - 1$ critical points of $\gamma^{(2)}$. Given $\varepsilon, \delta > 0$, the $\mathrm{lcss}_{\varepsilon,\delta}(\gamma^{(1)}, \gamma^{(2)})$ is defined as follows [95]:

$$\mathrm{lcss}_{\varepsilon,\delta}(\gamma^{(1)}, \gamma^{(2)}) = \begin{cases} 0, & \text{if } \gamma^{(1)} \text{ or } \gamma^{(2)} \text{ is empty} \\ 1 + \mathrm{lcss}_{\varepsilon,\delta}(\gamma^{(1)}, \gamma^{(2)}), & \text{if } \|c^{(1)}_{k_1} - c^{(2)}_{k_2}\| < \varepsilon \text{ and } |k_1 - k_2| < \delta \\ \max\big(\mathrm{lcss}_{\varepsilon,\delta}(\mathrm{Head}(\gamma^{(1)}), \gamma^{(2)}), \mathrm{lcss}_{\varepsilon,\delta}(\gamma^{(1)}, \mathrm{Head}(\gamma^{(2)}))\big), & \text{otherwise} \end{cases}$$

LCSS distance is defined as $\mathrm{LCSS}_{\varepsilon,\delta}(\gamma^{(1)}, \gamma^{(2)}) = 1 - \frac{\mathrm{lcss}_{\varepsilon,\delta}(\gamma^{(1)}, \gamma^{(2)})}{\max(k_1, k_2)}$.

Edit Distance for Real Sequences. It is similar to the edit distance on strings, and seeking the minimum number of edit operations required to change one trajectory to another

[23]. For EDR with $\varepsilon > 0$, $\gamma^{(1)}$ and $\gamma^{(2)}$ are considered to be the same if $k_1 = k_2$ and $\|c_i^{(1)} - c_i^{(2)}\| < \varepsilon$.

Locality Sensitive Hashing Distance. Given a point set $Q \subset \mathbb{R}^2$, and $r > 0$, It considers the disks with centers in Q and radius equal to r. For LSH1$_Q$, each trajectory is converted to a bit vector of length $|Q|$, and each bit represents the intersection of the trajectory with a disk. and uses Hamming distance of two bit vectors to define the distance between two curves. For LSH2$_Q$, each trajectories is converted to a sequence representing the order in which the trajectory enters and exits the disks, and uses edit distance of two sequence to define the distance between two curves [13].

B.3 The error of LCSS, EDR and LSH with Other Parameters in Section 4.4

In the experiment of Section 4.4, the computation of LCSS, EDR, LSH1$_Q$ and LSH2$_Q$ involves some parameters, and we only give the result of best parameter for these distances. In this section, we describe the change of error statistics for these distances with different parameters, and show how we obtain the best parameter in each experiment. We use bold font to mark the smallest mean error and the corresponding median and standard deviation (SD). In practice, a user would either guess a set of parameters which may be suboptimal, or they need to do an expensive parameter search (on a held out set) to choose parameters, which could greatly increase the runtime. In machine learning, the best parameter is usually chosen through cross-validation only on training data. However, for simplicity, in this work we directly choose the parameter that can yield the best final result of the experiment, which

Table B.1: Mean error of LCSS in Table 4.1 with different parameters.

mean ε / δ	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.1115	0.0822	0.0856	0.0969	0.1105	0.1297	0.1532	0.1749	0.1902	0.2003	0.2087	0.2182
2	0.0940	0.0785	0.0840	0.0954	0.1085	0.1278	0.1511	0.1731	0.1879	0.1987	0.2064	0.2164
3	0.0901	0.0769	0.0833	0.0946	0.1078	0.1271	0.1503	0.1718	0.1865	0.1977	0.2057	0.2160
4	0.0860	0.0755	0.0823	0.0936	0.1077	0.1267	0.1496	0.1707	0.1861	0.1966	0.2050	0.2151
5	0.0846	0.0745	0.0819	0.0935	0.1079	0.1269	0.1495	0.1704	0.1857	0.1961	0.2046	0.2150
6	0.0826	0.0739	0.0821	0.0939	0.1079	0.1265	0.1494	0.1706	0.1855	0.1958	0.2045	0.2149
7	0.0816	0.0734	0.0823	0.0937	0.1078	0.1261	0.1490	0.1702	0.1853	0.1957	0.2043	0.2147
8	0.0802	0.0729	0.0817	0.0935	0.1075	0.1261	0.1489	0.1702	0.1852	0.1957	0.2041	0.2145
9	0.0795	0.0721	0.0815	0.0933	0.1075	0.1262	0.1490	0.1699	0.1849	0.1955	0.2039	0.2142
10	0.0783	**0.0714**	0.0811	0.0930	0.1072	0.1258	0.1485	0.1695	0.1845	0.1951	0.2037	0.2140

involves the test data. Obviously, this is a help for distances that need parameter tuning.

Table B.2: Median error of LCSS in Table 4.1 with different parameters.

mean δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.0869	0.0577	0.0589	0.0652	0.0741	0.0889	0.1029	0.1143	0.1240	0.1325	0.1387	0.1474
2	0.0707	0.0536	0.0576	0.0643	0.0722	0.0868	0.1000	0.1118	0.1222	0.1304	0.1360	0.1458
3	0.0667	0.0531	0.0571	0.0640	0.0720	0.0867	0.1000	0.1103	0.1200	0.1297	0.1357	0.1464
4	0.0625	0.0526	0.0565	0.0640	0.0720	0.0864	0.1000	0.1094	0.1200	0.1278	0.1353	0.1449
5	0.0608	0.0524	0.0564	0.0643	0.0728	0.0865	0.1000	0.1087	0.1194	0.1274	0.1357	0.1449
6	0.0590	0.0516	0.0567	0.0649	0.0729	0.0857	0.1000	0.1088	0.1189	0.1267	0.1353	0.1449
7	0.0583	0.0512	0.0571	0.0647	0.0728	0.0857	0.0987	0.1088	0.1187	0.1267	0.1353	0.1446
8	0.0571	0.0506	0.0568	0.0646	0.0730	0.0857	0.0984	0.1083	0.1187	0.1266	0.1346	0.1444
9	0.0566	0.0500	0.0564	0.0645	0.0731	0.0857	0.0984	0.1079	0.1182	0.1261	0.1340	0.1444
10	0.0556	**0.0500**	0.0563	0.0643	0.0728	0.0850	0.0976	0.1076	0.1179	0.1256	0.1336	0.1440

Table B.3: Error standard deviation of LCSS in Table 4.1 with different parameters.

SD δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.0930	0.0839	0.0879	0.0994	0.1129	0.1273	0.1484	0.1673	0.1775	0.1834	0.1871	0.1916
2	0.0857	0.0823	0.0866	0.0979	0.1109	0.1267	0.1477	0.1660	0.1764	0.1830	0.1867	0.1911
3	0.0846	0.0797	0.0861	0.0969	0.1102	0.1263	0.1471	0.1655	0.1760	0.1826	0.1865	0.1909
4	0.0826	0.0773	0.0847	0.0957	0.1098	0.1259	0.1469	0.1653	0.1760	0.1822	0.1861	0.1904
5	0.0822	0.0764	0.0845	0.0953	0.1097	0.1258	0.1467	0.1652	0.1759	0.1821	0.1860	0.1904
6	0.0809	0.0759	0.0843	0.0951	0.1096	0.1258	0.1466	0.1652	0.1758	0.1822	0.1861	0.1905
7	0.0808	0.0757	0.0844	0.0948	0.1095	0.1256	0.1464	0.1650	0.1756	0.1820	0.1860	0.1905
8	0.0793	0.0751	0.0838	0.0948	0.1094	0.1255	0.1465	0.1650	0.1755	0.1821	0.1861	0.1906
9	0.0792	0.0746	0.0838	0.0946	0.1094	0.1255	0.1465	0.1651	0.1756	0.1821	0.1861	0.1906
10	0.0777	**0.0738**	0.0834	0.0943	0.1092	0.1254	0.1464	0.1651	0.1754	0.1821	0.1861	0.1906

Table B.4: Classification Error of EDR in Table 4.1 with different parameters.

ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
mean	0.1070	**0.0802**	0.0846	0.0957	0.1096	0.1289	0.1521	0.1744	0.1894	0.1997	0.2078	0.2175
median	0.0833	**0.0554**	0.0581	0.0640	0.0731	0.0875	0.1009	0.1139	0.1229	0.1319	0.1378	0.1462
SD	0.0918	**0.0835**	0.0876	0.0990	0.1125	0.1272	0.1482	0.1671	0.1773	0.1834	0.1872	0.1916

Table B.5: Classification Error of $LSH1_Q$ and $LSH2_Q$ in Table 4.1 with different parameters.

r	0.0050	0.0100	0.0200	0.0300	0.0400	0.0500	0.0600	0.0700	0.0800	0.0900	0.1000	0.1100
mean	0.4145	0.3792	0.3143	0.2645	0.2197	0.1374	**0.1290**	0.1501	0.1487	0.1774	0.1680	0.1633
$LSH1_{Q_1}$ median	0.3913	0.3500	0.2693	0.2121	0.1667	0.1000	**0.0949**	0.1114	0.1046	0.1133	0.1154	0.1179
SD	0.2481	0.2303	0.2116	0.2010	0.1774	0.1236	**0.1130**	0.1297	0.1328	0.1691	0.1523	0.1440
mean	0.4161	0.3873	0.3449	0.3043	0.2798	0.2637	0.2574	0.2494	0.2445	0.2415	**0.2409**	0.2426
$LSH2_{Q_1}$ median	0.3919	0.3644	0.3154	0.2605	0.2333	0.2281	0.2255	0.2275	0.2216	0.2195	**0.2182**	0.2191
SD	0.2492	0.2304	0.2137	0.2013	0.1952	0.1736	0.1647	0.1496	0.1483	0.1450	**0.1450**	0.1467

Table B.6: Classification Error of $LSH1_Q$ and $LSH2_Q$ in Table 4.4 with different parameters.

r	0.0050	0.0100	0.0200	0.0300	0.0400	0.0500	0.0600	0.0700	0.0800	0.0900	0.1000	0.1100
mean	0.4113	0.3827	0.3239	0.2615	0.2303	0.2139	0.1830	0.1339	0.1151	0.1173	**0.1122**	0.1223
$LSH1_{Q_2}$ median	0.3862	0.3448	0.2835	0.2079	0.1650	0.1429	0.1169	0.0958	0.0884	0.0875	**0.0860**	0.0923
SD	0.2532	0.2482	0.2256	0.1969	0.1952	0.1930	0.1750	0.1228	0.1008	0.1053	**0.0964**	0.1040
mean	0.3653	0.2740	0.2107	0.1766	0.1894	0.1571	0.1457	0.1399	0.1275	0.1409	**0.1190**	0.1217
$LSH1_{Q_3}$ median	0.3333	0.2304	0.1639	0.1312	0.1299	0.1040	0.1000	0.0969	0.0889	0.0917	**0.0861**	0.0909
SD	0.2196	0.1930	0.1666	0.1514	0.1696	0.1534	0.1389	0.1333	0.1216	0.1395	**0.1110**	0.1066
mean	0.4150	0.3968	0.3482	0.3076	0.2860	0.2690	0.2516	0.2437	0.2335	0.2293	**0.2278**	0.2278
$LSH2_{Q_2}$ median	0.3889	0.3644	0.3064	0.2683	0.2467	0.2325	0.2190	0.2105	0.2060	0.2048	**0.2027**	0.2050
SD	0.2529	0.2484	0.2290	0.1984	0.1842	0.1757	0.1636	0.1585	0.1483	0.1427	**0.1408**	0.1396
mean	0.3748	0.3109	0.2701	0.2471	0.2324	0.2204	0.2129	**0.2070**	0.2133	0.2099	0.2089	0.2081
$LSH2_{Q_3}$ median	0.3424	0.2819	0.2345	0.2136	0.2083	0.2000	0.1921	**0.1858**	0.1935	0.1896	0.1893	0.1897
SD	0.2262	0.1895	0.1694	0.1581	0.1409	0.1340	0.1280	**0.1256**	0.1269	0.1254	0.1244	0.1215

Table B.7: Mean error of LCSS in Table 4.6 with different parameters.

mean ε / δ	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.2761	0.2956	0.2634	0.2676	0.2839	0.3046	0.3049	0.3147	0.3256	0.3459	0.3527	0.3582
2	0.3123	0.2761	0.2718	0.2647	0.2866	0.3148	0.3112	0.3160	0.3267	0.3458	0.3552	0.3598
3	0.3116	0.2905	0.2685	**0.2448**	0.2941	0.3350	0.3129	0.3160	0.3267	0.3458	0.3542	0.3598
4	0.3112	0.2911	0.2552	0.2556	0.2937	0.3347	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598
5	0.2823	0.2919	0.2680	0.2656	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598
6	0.2883	0.2904	0.2691	0.2726	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3596
7	0.2931	0.2887	0.2645	0.2726	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598
8	0.2952	0.2828	0.2655	0.2726	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598
19	0.2946	0.2831	0.2655	0.2726	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598
10	0.2934	0.2831	0.2655	0.2726	0.2965	0.3352	0.3135	0.3160	0.3267	0.3458	0.3542	0.3598

Table B.8: Median error of LCSS in Table 4.6 with different parameters.

median δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.2778	0.3056	0.2500	0.2778	0.2778	0.3056	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
2	0.3056	0.2778	0.2778	0.2500	0.2778	0.3056	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
3	0.3056	0.2778	0.2778	**0.2500**	0.2778	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
4	0.3056	0.2778	0.2500	0.2500	0.2778	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
5	0.2778	0.2778	0.2778	0.2500	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
6	0.2778	0.2778	0.2778	0.2778	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
7	0.3056	0.2778	0.2500	0.2778	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
8	0.3056	0.2778	0.2500	0.2778	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
9	0.3056	0.2778	0.2500	0.2778	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
10	0.3056	0.2778	0.2500	0.2778	0.3056	0.3333	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611

Table B.9: Error standard deviation of LCSS in Table 4.6 with different parameters.

SD δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.0543	0.0612	0.0618	0.0637	0.0597	0.0587	0.0502	0.0356	0.0277	0.0188	0.0127	0.0168
2	0.0582	0.0593	0.0619	0.0618	0.0597	0.0619	0.0507	0.0351	0.0272	0.0187	0.0149	0.0175
3	0.0545	0.0589	0.0618	**0.0605**	0.0608	0.0617	0.0504	0.0351	0.0272	0.0187	0.0139	0.0175
4	0.0546	0.0603	0.0592	0.0606	0.0599	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
5	0.0558	0.0588	0.0612	0.0621	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
6	0.0542	0.0572	0.0602	0.0628	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
7	0.0541	0.0581	0.0592	0.0629	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
8	0.0544	0.0558	0.0594	0.0629	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
9	0.0541	0.0558	0.0594	0.0629	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175
10	0.0540	0.0558	0.0594	0.0629	0.0589	0.0615	0.0503	0.0351	0.0272	0.0187	0.0139	0.0175

Table B.10: Classification error of EDR in Table 4.6 with different parameters.

ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
mean	0.2748	0.2932	0.2661	**0.2640**	0.2854	0.3036	0.3050	0.3147	0.3256	0.3459	0.3527	0.3582
median	0.2778	0.2778	0.2639	**0.2500**	0.2778	0.3056	0.3056	0.3056	0.3333	0.3611	0.3611	0.3611
SD	0.0531	0.0619	0.0606	**0.0622**	0.0591	0.0595	0.0501	0.0356	0.0277	0.0188	0.0127	0.0168

Table B.11: Classification error of LSH1_Q and LSH2_Q in Table 4.6 with different parameters.

	r	0.0050	0.0100	0.0200	0.0300	0.0400	0.0500	0.0600	0.0700	0.0800	0.0900	0.1000	0.1100
LSH1_{Q_1}	mean	0.3360	0.2767	**0.2673**	0.2784	0.3211	0.3804	0.3647	0.3707	0.3627	0.3616	0.3611	0.3659
	median	0.3611	0.2778	**0.2778**	0.2778	0.3333	0.3611	0.3611	0.3611	0.3611	0.3611	0.3611	0.3611
	SD	0.0356	0.0512	**0.0448**	0.0531	0.0618	0.0342	0.0261	0.0348	0.0139	0.0066	0.0000	0.0194
LSH2_{Q_1}	mean	0.3361	0.2959	0.2789	0.2997	0.2869	0.2830	0.2811	0.2543	**0.2516**	0.2619	0.2684	0.2834
	median	0.3611	0.3056	0.2778	0.3056	0.2778	0.2778	0.2778	0.2500	**0.2500**	0.2778	0.2778	0.2778
	SD	0.0355	0.0449	0.0395	0.0510	0.0515	0.0561	0.0468	0.0459	**0.0467**	0.0413	0.0400	0.0371
LSH1_{Q_2}	mean	0.3365	0.2517	0.2352	**0.2209**	0.2468	0.2642	0.3334	0.3020	0.3754	0.3668	0.3626	0.3611
	median	0.3333	0.2500	0.2222	**0.2222**	0.2500	0.2500	0.3333	0.3056	0.3611	0.3611	0.3611	0.3611
	SD	0.0265	0.0505	0.0611	**0.0622**	0.0590	0.0522	0.0525	0.0486	0.0298	0.0221	0.0191	0.0000
LSH2_{Q_2}	mean	0.3472	0.3480	0.3428	0.2879	0.3131	**0.2690**	0.2945	0.2857	0.3217	0.3072	0.3249	0.3164
	median	0.3611	0.3611	0.3333	0.2778	0.3056	**0.2778**	0.3056	0.2778	0.3333	0.3056	0.3333	0.3056
	SD	0.0288	0.0254	0.0471	0.0430	0.0528	**0.0464**	0.0475	0.0481	0.0384	0.0464	0.0466	0.0387

Table B.12: Mean error of LCSS in Table 4.7 with different parameters.

mean δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.4961	0.4776	0.4484	0.4584	0.4068	0.4395	0.4412	0.4033	0.4233	0.4585	0.5073	0.5142
2	0.4112	0.3867	0.4405	0.4520	0.4363	0.4539	0.4238	0.4611	0.4999	0.5007	0.5062	0.5299
3	0.4025	0.4163	0.4903	0.4728	0.4343	0.4307	0.4389	0.4448	0.4656	0.4737	0.4814	0.5158
4	0.3504	0.4115	0.4320	0.4481	0.4435	0.4066	0.4319	0.4546	0.4397	0.4511	0.4631	0.4901
5	0.3509	0.4190	0.4082	0.4217	0.4177	0.4061	0.4378	0.4453	0.4606	0.4389	0.4738	0.4983
6	0.3481	0.4117	0.3961	0.4000	0.3939	0.4045	0.4368	0.4465	0.4592	0.4391	0.4759	0.4993
7	0.3527	0.4241	0.3996	0.4009	0.3947	0.4071	0.4308	0.4387	0.4569	0.4397	0.4780	0.4993
8	**0.3437**	0.4141	0.3998	0.4009	0.3947	0.4064	0.4308	0.4324	0.4554	0.4390	0.4780	0.4993
9	0.3499	0.4244	0.4039	0.3969	0.3961	0.4064	0.4308	0.4324	0.4554	0.4390	0.4780	0.4993
10	0.3582	0.4329	0.4041	0.3969	0.3961	0.4064	0.4308	0.4324	0.4554	0.4390	0.4780	0.4993

Table B.13: Median error of LCSS in Table 4.7 with different parameters.

median δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.5000	0.5000	0.4444	0.4444	0.3889	0.4444	0.4444	0.3889	0.4444	0.4444	0.5000	0.5000
2	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000	0.5000	0.5000
3	0.3889	0.4444	0.5000	0.4444	0.4444	0.4444	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
4	0.3333	0.3889	0.4444	0.4444	0.4444	0.3889	0.4444	0.4444	0.4444	0.4444	0.4444	0.5000
5	0.3333	0.4444	0.3889	0.4444	0.4444	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
6	0.3333	0.3889	0.3889	0.3889	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
7	0.3333	0.4444	0.3889	0.3889	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
8	**0.3333**	0.4167	0.3889	0.3889	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
9	0.3333	0.4444	0.3889	0.3889	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000
10	0.3333	0.4444	0.3889	0.3889	0.3889	0.3889	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000

Table B.14: Error standard deviation of LCSS in Table 4.7 with different parameters.

SD δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
1	0.0385	0.0596	0.0798	0.0784	0.0836	0.0824	0.0815	0.0760	0.0780	0.0827	0.0874	0.0879
2	0.0715	0.0791	0.0812	0.0836	0.0861	0.0787	0.0830	0.0829	0.0833	0.0853	0.0862	0.0900
3	0.0730	0.0760	0.0843	0.0836	0.0824	0.0749	0.0833	0.0828	0.0886	0.0917	0.0888	0.0963
4	0.0760	0.0807	0.0857	0.0851	0.0852	0.0759	0.0860	0.0850	0.0882	0.0893	0.0875	0.0951
5	0.0832	0.0884	0.0879	0.0855	0.0797	0.0766	0.0850	0.0847	0.0869	0.0886	0.0880	0.0951
6	0.0835	0.0864	0.0878	0.0866	0.0818	0.0769	0.0859	0.0834	0.0849	0.0885	0.0879	0.0951
7	0.0812	0.0848	0.0873	0.0888	0.0838	0.0761	0.0851	0.0822	0.0834	0.0880	0.0874	0.0951
8	**0.0812**	0.0860	0.0874	0.0888	0.0838	0.0758	0.0851	0.0793	0.0829	0.0880	0.0874	0.0951
9	0.0814	0.0848	0.0873	0.0886	0.0827	0.0758	0.0851	0.0793	0.0829	0.0880	0.0874	0.0951
10	0.0811	0.0839	0.0871	0.0886	0.0827	0.0758	0.0851	0.0793	0.0829	0.0880	0.0874	0.0951

Table B.15: Classification error of EDR in Table 4.7 with different parameters.

ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
mean	0.4632	0.4541	0.4171	0.4450	**0.3916**	0.4134	0.4422	0.4259	0.4618	0.4559	0.4681	0.5123
median	0.4444	0.4444	0.4444	0.4444	**0.3889**	0.3889	0.4444	0.4444	0.4444	0.4444	0.4444	0.5000
SD	0.0512	0.0654	0.0806	0.0824	**0.0823**	0.0801	0.0761	0.0775	0.0786	0.0890	0.0899	0.0934

Table B.16: Classification Error of LSH1_Q and LSH2_Q in Table 4.7 with different parameters.

	r	0.0050	0.0100	0.0200	0.0300	0.0400	0.0500	0.0600	0.0700	0.0800	0.0900	0.1000	0.1100
LSH1_Q	mean	0.5098	**0.2524**	0.2950	0.4878	0.4443	0.4691	0.4494	0.4558	0.5046	0.5103	0.4439	0.4305
	median	0.5000	**0.2222**	0.2778	0.5000	0.4444	0.4444	0.4444	0.4444	0.5000	0.5000	0.4444	0.4444
	SD	0.0240	**0.0990**	0.0818	0.0802	0.0789	0.0813	0.0922	0.0768	0.0671	0.0822	0.0828	0.0779
LSH2_Q	mean	0.5000	0.4547	**0.3248**	0.3850	0.5271	0.5400	0.5216	0.5130	0.4828	0.4943	0.4406	0.4865
	median	0.5000	0.4444	**0.3333**	0.3889	0.5278	0.5556	0.5000	0.5000	0.5000	0.5000	0.4444	0.5000
	SD	0	0.0863	**0.0916**	0.0872	0.0825	0.0703	0.0677	0.0848	0.0723	0.0701	0.0872	0.0839

B.4 Choose at Most 40 Points for Each Trajectory of Beijing Drivers

In this section, we redo the Beijing drivers experiment in Section 4.4.3, and the only difference is when a trajectory in the raw data has at least 10 and at most 40 critical points we directly retain all the critical points of this trajectory, and when a trajectory in the raw data has more than 40 critical points, we use Douglas-Peucker algorithm to convert it to a trajectory with 40 critical points. So, in the data after preprocessing, trajectories have different numbers of critical points, which implies Euclidean distance and $\mathtt{d}_Q^{\leftrightarrow}$ cannot be used in this case. We use the same points set Q_1, Q_2, Q_3 and $\tilde{Q}_1$, $\tilde{Q}_2$, $\tilde{Q}_3$ in Section 4.4.3.

The running result for different algorithms and distances is shown in Table B.17 and Table B.18. From these two tables, we can see the performances of most distances are slightly improved, except LCSS whose mean error is greatly reduced and achieves the best result. However, $\mathtt{d}_Q$ and $\mathtt{d}_Q^{\pi}$ are still competitive. $\mathtt{d}_Q^{\pi}$ achieves the best result in all distances except LCSS. Moreover, LCSS needs two parameters and for each pair of parameter (δ, ε) it takes more than 6 hours to get the result of error statistics (we use a computer with Intel Xeon(R) CPU E5-2660 v3 @2.6GHz (2 processors), 6GB RAM and Windows 10 operating system). So, to choose the best parameters for LCSS we need to try many or all choices, and this is why LCSS is actually dramatically slower! For example, the computation of all errors in Table B.19 needs more than 720 hours. By contrast, $\mathtt{d}_Q^{\pi}$ does not need parameters and it only takes about 7 minutes to get the result of error statistics for the case $|Q| = 200$. It takes about 10 minutes to convert all trajectories to vectors in $\mathbb{R}^{400}$. So, the computation of $\mathtt{d}_Q^{\pi}$ is much faster than LCSS (17 minutes vs. 720 hours), even if the time of data preprocessing is considered. The computation of $\mathtt{d}_Q$ is faster than $\mathtt{d}_Q^{\pi}$, since for $\mathtt{d}_Q$ each trajectory is converted to a vector in $\mathbb{R}^{200}$ rather than $\mathbb{R}^{400}$. Other distances, except Euclidean distance,

Table B.17: Classification error on Beijing Drivers ($|Q| = 20$, each trajectory contains at most 40 critical points)

	distance	mean	median	SD
KNN	$\mathtt{d}_{Q_1}$	0.0823	0.0676	0.0622
	$\mathtt{d}_{Q_2}$	0.0820	0.0667	0.0621
	$\mathtt{d}_{Q_3}$	0.0804	0.0655	0.0620
	$\mathtt{d}^{\pi}_{Q_1}$	0.0724	0.0583	0.0575
	$\mathtt{d}^{\pi}_{Q_2}$	0.0720	0.0571	0.0571
	$\mathtt{d}^{\pi}_{Q_3}$	0.0698	0.0556	0.0565
	dF	0.1045	0.0875	0.0731
	DTW	0.0708	0.0556	0.0594
	dH	0.0889	0.0733	0.0652
	LCSS ($\varepsilon = 0.001, \delta = 40$)	**0.0651**	0.0450	0.0695
	EDR($\varepsilon = 0.005$)	0.0756	0.0533	0.0762
	LSH1_{Q_1} (r=0.06)	0.1261	0.0927	0.1116
	LSH1_{Q_2} (r=0.1)	0.1132	0.0869	0.0969
	LSH1_{Q_3} (r=0.1)	0.1182	0.0867	0.1082
	LSH2_{Q_1} (r=0.1)	0.2447	0.2250	0.1423
	LSH2_{Q_2} (r=0.1)	0.2527	0.2308	0.1513
	LSH2_{Q_3} (r=0.11)	0.2113	0.1944	0.1185
linear SVM	$\mathtt{d}_{Q_1}$	0.2069	0.1854	0.1257
	$\mathtt{d}_{Q_2}$	0.2042	0.1818	0.1249
	$\mathtt{d}_{Q_3}$	0.2042	0.1818	0.1246
	$\mathtt{d}^{\pi}_{Q_1}$	0.2047	0.1892	0.1219
	$\mathtt{d}^{\pi}_{Q_2}$	0.2005	0.1818	0.1213
	$\mathtt{d}^{\pi}_{Q_3}$	0.2019	0.1833	0.1213
quadratic SVM	$\mathtt{d}_{Q_1}$	0.2197	0.1778	0.1685
	$\mathtt{d}_{Q_2}$	0.2155	0.1702	0.1695
	$\mathtt{d}_{Q_3}$	0.2155	0.1715	0.1676
	$\mathtt{d}^{\pi}_{Q_1}$	0.1998	0.1468	0.1685
	$\mathtt{d}^{\pi}_{Q_2}$	0.2010	0.1480	0.1688
	$\mathtt{d}^{\pi}_{Q_3}$	0.1942	0.1395	0.1676
Gaussian SVM	$\mathtt{d}_{Q_1}$	0.0730	0.0588	0.0594
	$\mathtt{d}_{Q_2}$	0.0740	0.0599	0.0593
	$\mathtt{d}_{Q_3}$	0.0731	0.0587	0.0593
	$\mathtt{d}^{\pi}_{Q_1}$	0.0737	0.0594	0.0597
	$\mathtt{d}^{\pi}_{Q_2}$	0.0738	0.0596	0.0595
	$\mathtt{d}^{\pi}_{Q_3}$	0.0725	0.0581	0.0593

are also slower than $\mathtt{d}_Q$ and $\mathtt{d}^{\pi}_Q$: dF: about 9.2 hours, dH: about 4.2 hours, DTW: about 7.1 hours, EDR: about 1 hour for one parameter on average, LSH1_{Q_1}: about 3 minutes for one parameter on average, LSH2_{Q_1}: about 28 minutes for one parameter on average.

The change of error statistics for LCSS, EDR, LSH1_Q and LSH2_Q with different parameters are shown in Tables B.19, B.20, B.21, B.22 and B.23. The smallest mean error and the corresponding median and standard deviation (SD) are marked by bold font.

Table B.18: Classification error on Beijing Drivers ($|Q| = 200$, each trajectory contains at most 40 critical points)

	distance	mean	median	SD
KNN	$\mathtt{d}_{\tilde{Q}_1}$	0.0810	0.0654	0.0626
	$\mathtt{d}_{\tilde{Q}_2}$	0.0811	0.0656	0.0619
	$\mathtt{d}_{\tilde{Q}_3}$	0.0805	0.0651	0.0623
	$\mathtt{d}^{\pi}_{\tilde{Q}_1}$	0.0707	0.0558	0.0575
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.0708	0.0563	0.0569
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.0699	0.0556	0.0566
linear SVM	$\mathtt{d}_{\tilde{Q}_1}$	0.1422	0.1129	0.1057
	$\mathtt{d}_{\tilde{Q}_2}$	0.1426	0.1126	0.1059
	$\mathtt{d}_{\tilde{Q}_3}$	0.1416	0.1125	0.1053
	$\mathtt{d}^{\pi}_{\tilde{Q}_1}$	0.1429	0.1185	0.1014
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.1438	0.1190	0.1020
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.1427	0.1179	0.1014
quadratic SVM	$\mathtt{d}_{\tilde{Q}_1}$	0.1391	0.0909	0.1410
	$\mathtt{d}_{\tilde{Q}_2}$	0.1413	0.0913	0.1435
	$\mathtt{d}_{\tilde{Q}_3}$	0.1390	0.0900	0.1418
	$\mathtt{d}^{\pi}_{\tilde{Q}_1}$	0.2597	0.2222	0.1912
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.2609	0.2208	0.1928
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.2590	0.2181	0.1917
Gaussian SVM	$\mathtt{d}_{\tilde{Q}_1}$	0.0723	0.0582	0.0589
	$\mathtt{d}_{\tilde{Q}_2}$	0.0725	0.0582	0.0587
	$\mathtt{d}_{\tilde{Q}_3}$	0.0719	0.0577	0.0585
	$\mathtt{d}^{\pi}_{\tilde{Q}_1}$	0.0728	0.0586	0.0596
	$\mathtt{d}^{\pi}_{\tilde{Q}_2}$	0.0728	0.0583	0.0596
	$\mathtt{d}^{\pi}_{\tilde{Q}_3}$	0.0724	0.0580	0.0592

Table B.19: Mean error of LCSS in Table B.17 with different parameters.

mean δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
31	0.0666	0.0669	0.0760	0.0867	0.0996	0.1152	0.1345	0.1536	0.1680	0.1775	0.1859	0.1947
32	0.0665	0.0666	0.0760	0.0868	0.0994	0.1151	0.1344	0.1535	0.1678	0.1774	0.1857	0.1947
33	0.0664	0.0666	0.0759	0.0868	0.0994	0.1151	0.1344	0.1535	0.1678	0.1775	0.1858	0.1947
34	0.0664	0.0665	0.0759	0.0867	0.0994	0.1151	0.1344	0.1535	0.1678	0.1775	0.1857	0.1948
35	0.0660	0.0664	0.0759	0.0867	0.0995	0.1152	0.1344	0.1535	0.1678	0.1775	0.1857	0.1947
36	0.0657	0.0661	0.0758	0.0867	0.0994	0.1151	0.1344	0.1535	0.1678	0.1775	0.1857	0.1947
37	0.0653	0.0661	0.0757	0.0867	0.0993	0.1151	0.1344	0.1535	0.1678	0.1774	0.1857	0.1946
38	0.0652	0.0660	0.0755	0.0865	0.0993	0.1150	0.1343	0.1534	0.1677	0.1774	0.1856	0.1946
39	0.0652	0.0658	0.0755	0.0864	0.0993	0.1150	0.1343	0.1534	0.1677	0.1774	0.1856	0.1946
40	**0.0651**	0.0658	0.0754	0.0863	0.0992	0.1149	0.1343	0.1534	0.1677	0.1773	0.1856	0.1945

Table B.20: Median error of LCSS in Table B.17 with different parameters.

mean δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
31	0.0459	0.0477	0.0547	0.0617	0.0700	0.0809	0.0912	0.1000	0.1101	0.1174	0.1242	0.1332
32	0.0459	0.0474	0.0548	0.0617	0.0700	0.0809	0.0911	0.1000	0.1100	0.1174	0.1242	0.1333
33	0.0460	0.0474	0.0547	0.0617	0.0700	0.0809	0.0911	0.1000	0.1100	0.1174	0.1242	0.1332
34	0.0459	0.0474	0.0547	0.0617	0.0700	0.0809	0.0911	0.1000	0.1100	0.1174	0.1240	0.1333
35	0.0457	0.0474	0.0547	0.0617	0.0700	0.0809	0.0911	0.1000	0.1100	0.1174	0.1242	0.1333
36	0.0455	0.0471	0.0545	0.0617	0.0700	0.0808	0.0911	0.1000	0.1100	0.1174	0.1243	0.1324
37	0.0452	0.0471	0.0545	0.0617	0.0698	0.0808	0.0911	0.1000	0.1099	0.1174	0.1243	0.1323
38	0.0450	0.0469	0.0544	0.0616	0.0697	0.0808	0.0909	0.1000	0.1098	0.1174	0.1240	0.1323
39	0.0450	0.0467	0.0544	0.0615	0.0696	0.0808	0.0909	0.1000	0.1100	0.1174	0.1240	0.1323
40	**0.0450**	0.0467	0.0543	0.0615	0.0696	0.0808	0.0909	0.1000	0.1100	0.1174	0.1240	0.1322

Table B.21: Error standard deviation of LCSS in Table B.17 with different parameters.

SD δ \ ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
31	0.0705	0.0683	0.0759	0.0852	0.0984	0.1107	0.1297	0.1474	0.1587	0.1646	0.1685	0.1724
32	0.0706	0.0681	0.0758	0.0852	0.0983	0.1107	0.1297	0.1474	0.1587	0.1646	0.1685	0.1724
33	0.0703	0.0680	0.0757	0.0853	0.0983	0.1107	0.1297	0.1474	0.1587	0.1645	0.1685	0.1724
34	0.0703	0.0680	0.0756	0.0852	0.0983	0.1106	0.1296	0.1474	0.1587	0.1645	0.1685	0.1724
35	0.0699	0.0680	0.0757	0.0852	0.0983	0.1106	0.1296	0.1474	0.1587	0.1645	0.1684	0.1724
36	0.0698	0.0677	0.0756	0.0852	0.0982	0.1106	0.1296	0.1474	0.1587	0.1645	0.1684	0.1724
37	0.0696	0.0678	0.0757	0.0852	0.0982	0.1106	0.1296	0.1473	0.1587	0.1645	0.1684	0.1724
38	0.0695	0.0675	0.0755	0.0851	0.0982	0.1106	0.1296	0.1473	0.1587	0.1645	0.1684	0.1724
39	0.0695	0.0675	0.0755	0.0850	0.0982	0.1106	0.1296	0.1473	0.1586	0.1645	0.1684	0.1724
40	**0.0695**	0.0674	0.0754	0.0850	0.0982	0.1106	0.1296	0.1473	0.1586	0.1645	0.1684	0.1724

Table B.22: Classification Error of EDR in Table B.17 with different parameters.

ε	0.0010	0.0050	0.0100	0.0150	0.0200	0.0250	0.0300	0.0350	0.0400	0.0450	0.0500	0.0550
mean	0.0874	**0.0756**	0.0798	0.0909	0.1031	0.1188	0.1388	0.1590	0.1731	0.1824	0.1904	0.1993
median	0.0667	**0.0533**	0.0571	0.0643	0.0718	0.0833	0.0947	0.1062	0.1144	0.1220	0.1286	0.1368
SD	0.0826	**0.0762**	0.0785	0.0888	0.1004	0.1126	0.1322	0.1497	0.1605	0.1659	0.1698	0.1740

Table B.23: Classification Error of $LSH1_Q$ and $LSH2_Q$ in Table B.17 with different parameters.

r	0.0050	0.0100	0.0200	0.0300	0.0400	0.0500	0.0600	0.0700	0.0800	0.0900	0.1000	0.1100
mean	0.4140	0.3790	0.3099	0.2631	0.2172	0.1354	**0.1261**	0.1518	0.1476	0.1769	0.1673	0.1634
$LSH1_{Q_1}$ median	0.3914	0.3478	0.2632	0.2096	0.1667	0.1000	**0.0927**	0.1121	0.1037	0.1148	0.1149	0.1178
SD	0.2481	0.2312	0.2122	0.2007	0.1753	0.1218	**0.1116**	0.1321	0.1313	0.1667	0.1526	0.1441
mean	0.4094	0.3806	0.3201	0.2574	0.2272	0.2132	0.1823	0.1329	0.1154	0.1196	**0.1132**	0.1243
$LSH1_{Q_2}$ median	0.3819	0.3436	0.2757	0.2000	0.1607	0.1426	0.1174	0.0953	0.0886	0.0900	**0.0869**	0.0941
SD	0.2520	0.2466	0.2221	0.1971	0.1936	0.1931	0.1743	0.1214	0.1002	0.1069	**0.0969**	0.1042
mean	0.3640	0.2706	0.2084	0.1767	0.1876	0.1561	0.1453	0.1380	0.1274	0.1407	**0.1182**	0.1208
$LSH1_{Q_3}$ median	0.3333	0.2253	0.1606	0.1305	0.1273	0.1020	0.0987	0.0954	0.0889	0.0924	**0.0867**	0.0900
SD	0.2184	0.1925	0.1667	0.1523	0.1690	0.1525	0.1382	0.1317	0.1214	0.1374	**0.1082**	0.1057
mean	0.4161	0.3887	0.3482	0.3105	0.2963	0.2810	0.2723	0.2561	0.2470	0.2502	**0.2447**	0.2454
$LSH2_{Q_1}$ median	0.3939	0.3660	0.3231	0.2730	0.2552	0.2475	0.2401	0.2333	0.2276	0.2316	**0.2250**	0.2250
SD	0.2496	0.2315	0.2135	0.2027	0.1959	0.1772	0.1683	0.1501	0.1443	0.1435	**0.1423**	0.1427
mean	0.4138	0.3962	0.3594	0.3193	0.2986	0.2900	0.2666	0.2579	0.2560	0.2542	**0.2527**	0.2530
$LSH2_{Q_2}$ median	0.3871	0.3625	0.3196	0.2857	0.2641	0.2549	0.2369	0.2302	0.2321	0.2304	**0.2308**	0.2308
SD	0.2516	0.2480	0.2334	0.2038	0.1907	0.1819	0.1701	0.1618	0.1562	0.1545	**0.1513**	0.1498
mean	0.3761	0.3106	0.2681	0.2529	0.2384	0.2307	0.2228	0.2178	0.2174	0.2150	0.2125	**0.2113**
$LSH2_{Q_3}$ median	0.3478	0.2828	0.2363	0.2200	0.2138	0.2091	0.2042	0.2000	0.2010	0.2000	0.1960	**0.1944**
SD	0.2263	0.1864	0.1681	0.1576	0.1411	0.1324	0.1278	0.1256	0.1228	0.1221	0.1212	**0.1185**

REFERENCES

[1] A. Abdullah, S. Daruki, and J. M. Phillips, *Range counting coresets for uncertain data*, in SOCG, 2013.

[2] P. K. Agarwal, B. Aronov, S. Har-Peled, J. M. Phillips, K. Yi, and W. Zhang, *Nearest-neighbor searching under uncertainty II*, in PODS, 2013.

[3] P. K. Agarwal, S.-W. Cheng, Y. Tao, and K. Yi, *Indexing uncertain data*, in PODS, 2009.

[4] P. K. Agarwal, A. Efrat, S. Sankararaman, and W. Zhang, *Nearest-neighbor searching under uncertainty*, in PODS, 2012.

[5] P. K. Agarwal, S. Har-Peled, S. Suri, H. Yildiz, and W. Zhang, *Convex hulls under uncertainty*, in ESA, 2014.

[6] P. Agrawal, O. Benjelloun, A. D. Sarma, C. Hayworth, S. Nabar, T. Sugihara, and J. Widom, *Trio: A system for data, uncertainty, and lineage*, in PODS, 2006.

[7] G. Aloupis, *Geometric measures of data depth*, in Data Depth: Robust Multivariate Analysis, Computational Geometry and Applications, AMS, 2006.

[8] H. Alt and L. J. Guibas, *Discrete geometric shapes: Matching, interpolation, and approximation: A survey*, in Handbook of Computational Geometry, -, 1996.

[9] H. Alt, K. Mehlhorn, H. Wagener, and E. Welzl, *Congruence, similarity, and symmetries of geometric objects*, Discrete & Computational Geometry, 3 (1988), pp. 237–256.

[10] N. Amenta, S. Choi, and R. K. Kolluri, *The power crust*, in Proceedings of the sixth ACM symposium on Solid modeling and applications, 2001.

[11] M. Anthony and P. L. Bartlett, *Neural Network Learning: Theoretical Foundations*, Cambridge University Press, 1999.

[12] S. Arora, P. Raghavan, and S. Rao, *Approximation schemes for euclidean k-medians and related problems*, in STOC, 1998.

[13] M. Astefanoaei, P. Cesaretti, P. Katsikouli, and M. G. andRik Sarkar, *Multi-resolution sketches and locality sensitive hashing for fast trajectory processing*, in SIGSPATIAL, 2018.

[14] P. Bose, A. Maheshwari, and P. Morin, *Fast approximations for sums of distances clustering and the Fermet-Weber problem*, CGTA, 24 (2003), pp. 135–146.

[15] C. Boutsidis, M. W. Mahoney, and P. Drineas, *An improved approximation algorithm for the column subset selection problem*, in Proceedings of the 20th Annual ACM-SIAM Symposium on Discrete Algorithms, 2009.

[16] V. Braverman, D. Feldman, and H. Lang, *New frameworks for offline and streaming coreset constructions*, arXiv preprint arXiv:1612.00889, 2016.

[17] R. Chandrasekaran and A. Tamir, *Algebraic optimization: The Fermet-Weber location problem*, Mathematical Programming, 46 (1990), pp. 219–224.

[18] F. Chazal and D. Cohen-Steiner, *Geometric inference*. `https://geometrica.saclay.inria.fr/team/Fred.Chazal/papers/GeomInference5.pdf`.

[19] F. Chazal, D. Cohen-Steiner, and Q. Mérigot, *Geometric inference for probability measures*, Foundations of Computational Mathematics, (2010), pp. 1–19.

[20] F. Chazal and A. Lieutier, *The "λ-medial axis"*, Graphical Models, 67 (2005), pp. 304–331.

[21] C. Chen, C. Yuan, and C. Chen, *Solving M-modes using heuristic search*, in 25th International Joint Conference on Artificial Intelligence, 2016.

[22] D. Chen and J. M. Phillips, *Relative error embeddings for the gaussian kernel distance*, in Algorithmic Learning Theory, 2017.

[23] L. Chen, M. T. Özsu, and V. Oria, *Robust and fast similarity search for moving object trajectories.*, in SIGMOD, 2005.

[24] R. Cheng, Y. Xia, S. Prabhakar, R. Shah, and J. S. Vitter, *Efficient indexing methods for probabilistic threshold queries over uncertain data*, in VLDB, 2004.

[25] M. B. Cohen, C. Musco, and C. Musco, *Input sparsity time low-rank approximation via ridge leverage score sampling*, in ACM-SIAM Symposium on Discrete Algorithms, 2017.

[26] M. B. Cohen, C. Musco, and J. Pachocki., *Online row sampling*, in International Workshop on Approximation, Randomization, and Combinatorial Optimization, 2016.

[27] G. Cormode and M. Garafalakis, *Histograms and wavelets of probabilitic data*, in ICDE, 2009.

[28] G. Cormode and A. McGregor, *Approximation algorithms for clustering uncertain data*, in PODS, 2008.

[29] M. O. Cruz, H. Macedo, R. Barreto, and A. Guimaraes, *GPS Trajectories Data Set*, February 2016.

[30] N. Dalvi and D. Suciu, *Efficient query evaluation on probabilistic databases*, in VLDB, 2004.

[31] N. N. Dalvi, C. Ré, and D. Suciu, *Probabilistic databases: Diamonds in the dirt*, Comm. ACM, 52 (2009), pp. 86–94.

[32] P. Davies and U. Gather, *The breakdown point: Examples and counterexamples*, REVSTAT – Statitical Journal, 5 (2007), pp. 1–17.

[33] D. Donoho and P. J. Huber, *The notion of a breakdown point*, in A Festschrift for Erich L. Lehmann, P. Bickel, K. Doksum, and J. Hodges, eds., Wadsworth International Group, 1983, pp. 157–184.

[34] A. Driemel, J. M. Phillips, and I. Psarros, *On the vc dimension of metric balls under frechet and hausdorff distances*, in International Symposium on Computational Geometry, 2019.

[35] P. Drineas, M. Magdon-Ismail, M. W. Mahoney, and D. P. Woodruff, *Fast approximation of statistical leverage*, Journal of Machine Learning Research, 13 (2012), pp. 3475–3506.

[36] P. Drineas, M. W. Mahoney, and S. Muthukrishnan, *Relative-error CUR matrix decompositions*, SIAM Journal of MAtrix Analysis and Applications, 30 (2008), pp. 844–881.

[37] H. Edelsbrunner and E. P. Mücke, *Three-dimensional alpha shapes*, ACM Transactions on Graphics, 13 (1994), pp. 43–72.

[38] T. Eiter and H. Mannila, *Computing discrete Frechet distance*, tech. rep., Christian Doppler Laboratory for Expert Systems, 1994.

[39] Y. Fang, R. Cheng, W. Tang, S. Maniu, and X. S. Yang:, *Scalable algorithms for nearest-neighbor joins on big trajectory data*, in ICDE, 2016.

[40] D. Feldman and M. Langberg, *A unified framework for approximating and clustering data*, in Proceedings ACM Symposium on Theory of Computing, 2011.

[41] D. Feldman and M. Langberg, *A unified framework for approximating and clustering data*, in STOC, 2011, pp. 569–578.

[42] D. Feldman, M. Schmidt, and C. Sohler, *Turning big data into tiny data: Constant-size coresets for k-means, pca, and projective clustering*, in SODA, 2013, pp. 1434–1453.

[43] ——, *Turning big data into tiny data: Constant-size coresets for k-means, PCA, and projective clustering*, in Proceedings 24th ACM-SIAM Symposium on Discrete Algorithms, 2013.

[44] D. Feldman and L. J. Schulman, *Data reduction for weighted and outlier-resistant clustering*, in Proc. ACM-SIAM Symposium on Discrete Algorithms, 2012.

[45] D. C.-S. Frederic Chazal and A. Lieutier, *A sampling theory for compact sets in euclidean space*, DCG, 41 (2009), pp. 461–479.

[46] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis, *Nearest neighbor search on moving object trajectories*, in SSTD, 2005.

[47] E. G. Gilbert and C.-P. Foo, *Computing the distance between general convex objects in three-dimensional space.*, IEEE Transactions on Robotics and Automation, 6 (1990), pp. 53–61.

[48] E. G. Gilbert, D. W. Johnson, and S. S. Keerthi, *A fast procedurefor computing the distance between objects in three-dimensionalspace.*, IEEE J. Robotics and Automation, 4 (1988), pp. 193–203.

[49] T. F. Gonzalez, *Clustering to minimize the maximum intercluster distance*, Theoretical Computer Science, 38 (1985), pp. 293–306.

[50] R. H. Güting, T. Behr, and J. Xu, *Efficient k-nearest neighbor search on moving object trajectories*, in VLDB, 2010.

[51] F. Hampel, E. M. Ronchetti, P. J. Rousseeuw, and W. A. Stahel, *Robust Statistics: The Approach Based on Influence Functions*, Wiley, 1986.

[52] F. R. Hampel, *A general qualitative definition mof robustness*, Annals of Mathematical Statistics, 42 (1971), pp. 1887–1896.

[53] S. Har-Peled, *Geometric Approximation Algorithms*, Mathematical Surveys and Monographs, American Mathematical Society, 2011.

[54] X. He, D. G. Simplson, and S. L. Portnoy, *Breakdown robustness of tests*, Journal of the Maerican Statistical Association, 85 (1990), pp. 446–452.

[55] L. Huang and J. Li, *Approximating the expected values for combinatorial optimization problems over stochastic points*, in ICALP, 2015.

[56] P. J. Huber, *Robust Statistics*, Wiley, 1981.

[57] P. J. Huber and E. M. Ronchetti, *Breakdown point*, in Robust Statistics, John Wiley & Sons, Inc., 2009, p. 8.

[58] T. Jayram, A. McGregor, S. Muthukrishnan, and E. Vee, *Estimating statistical aggregates on probabilistic data streams*, ACM TODS, 33 (2008), pp. 1–30.

[59] W. B. Johnson and J. Lindenstrauss, *Extensions of Lipschitz maps into a Hilbert space*, Contemporary Mathematics, 26 (1984), pp. 189–206.

[60] A. G. Jørgensen, M. Löffler, and J. M. Phillips, *Geometric computation on indecisive points*, in WADS, 2011.

[61] S. Joshi, R. V. Kommaraju, J. M. Phillips, and S. Venkatasubramanian, *Comparing distributions and shapes using the kernel distance*, Proceedings 27th Annual Symposium on Computational Geometry, (2011).

[62] P. Kamousi, T. M. Chan, and S. Suri, *Stochastic minimum spanning trees in euclidean spaces*, in SOCG, 2011.

[63] M. Langberg and L. J. Schulman, *Universal ε-approximators for integrals*, in SODA, 2010, pp. 598–607.

[64] J. Li, B. Saha, and A. Deshpande, *A unified approach to ranking in probabilistic databases*, in VLDB, 2009.

[65] Y. Li, P. M. Long, and A. Srinivasan, *Improved bounds on the samples complexity of learning*, Journal of Computer and System Science, 62 (2001), pp. 516–527.

[66] D. Lin, R. Zhang, and A. Zhou, *Indexing fast moving objects for knn queries based on nearest landmarks*, GeoInformatica, 10 (2006), pp. 423–445.

[67] M. Löffler and J. Phillips, *Shape fitting on point sets with probability distributions*, in ESA, 2009.

[68] D. Lopaz-Paz, K. Muandet, B. Schölkopf, and I. Tolstikhin, *Towards a learning theory of cause-effect inference*, in International Conference on Machine Learning, 2015.

[69] H. P. Lopuhaa and P. J. Rousseeuw, *Breakdown points of affine equivaniant estimators of multivariate location and converiance matrices*, The Annals of Statistics, 19 (1991), pp. 229–248.

[70] S. M. Ma, Qiang and M. Sandler, *Frugal streaming for estimating quantiles: One (or two) memory suffices*, arXiv preprint arXiv: 1407.1121, (2014).

[71] F. Mémoli, *Gromov–Hausdorff distances in Euclidean spaces*, In Proc. non-rigid shape analysis and deformable image registration (NORDIA) workshop, (2008).

[72] M. Matheny, D. Xie, and J. M. Phillips, *Scalable spatial scan statistics for trajectories*, tech. rep., arXiv:1906.01693, 2019.

[73] K. Muandet, K. Fukumizu, B. Sriperumbudur, and B. Schölkopf, *Kernel mean embedding of distributions: A review and beyond*, Foundations and Trends in Machine Learning, 10 (2017), pp. 1–141.

[74] C. Musco and C. Musco, *Recursive sampling for the Nyström method*, in NIPS, 2017.

[75] J. M. Phillips and W. M. Tai, *Relative error rkhs embeddings for gaussian kernels*, tech. rep., arXiv:1811.04136, 2018.

[76] J. M. Phillips and P. Tang, *Simple distances for trajectories via landmarks*, tech. rep., arXiv:1804.11284, 2019.

[77] J. M. Phillips, B. Wang, and Y. Zheng, *Geomtric inference on kernel density estimates*, in SOCG, 2015.

[78] P. J. Rousseeuw, *Multivariate estimation with high breakdown point*, Mathematical Statistics and Applications, (1985), pp. 283–297.

[79] A. D. Sarma, O. Benjelloun, A. Halevy, S. Nabar, and J. Widom, *Representing uncertain data: models, properties, and algorithms*, VLDBJ, 18 (2009), pp. 989–1019.

[80] Z. Shang, G. Li, and Z. Bao, *Dita: Distributed in-memory trajectory analytics*, in SIGMOD, 2018.

[81] A. F. Siegel, *Robust regression using repeated medians*, Biometrika, 82 (1982), pp. 242–244.

[82] Y. Tao, R. Cheng, X. Xiao, W. K. Ngai, B. Kao, and S. Prabhakar, *Indexing multi-dimensional uncertain data with arbitrary probability density functions*, in VLDB, 2005.

[83] J. W. Tukey, *Mathematics and the picturing of data*, in Proceedings of the 1974 International Congress of Mathematics, Vancouver, vol. 2, 1975, pp. 523–531.

[84] M. van Kreveld and M. Löffler, *Largest bounding box, smallest diameter, and related problems on imprecise points*, CGTA, 43 (2010), pp. 419–433.

[85] V. Vapnik and A. Chervonenkis, *On the uniform convergence of relative frequencies of events to their probabilities*, Th. Probability and Applications, 16 (1971), pp. 264–280.

[86] K. Varadarajan and X. Xiao, *On the sensitivity of shape fitting problems*, in Proceedings International Conference on Foundations of Software Technology and Theoretical Computer Science, arxiv:1209.4893, 2012.

[87] M. Vlachos, G. Kollios, and D. Gunopulos, *Discovering similar multidimensional trajectories*, in ICDE, 2002.

[88] E. Weiszfeld, *Sur le point pour lequel la somme des distances de n points dennes est minimum*, Tohoku Mathmatics, 43 (1937), pp. 355–386.

[89] E. Weiszfeld and F. Plastria, *On the point for which the sum of the distances to n given points is minimum*, Annals of Operations Research, 167 (2009), pp. 7–41.

[90] A. H. Welsh, *The standard deviation*, in Aspects of Statistical Inference, Wiley-Interscience;, 1996, p. 245.

[91] F. Wu, Z. Li, W.-C. Lee, H. Wang, and Z. Huang, *Semantic annotation of mobility data using social media*, in WWW, 2015.

[92] D. Xie, F. Li, and J. M. Phillips, *Distributed trajectory similarity search*, in VLDB, 2017.

[93] B.-K. Yi, H. Jagadish, and C. Faloutsos, *Efficient retrieval of similar time sequences under time warping*, in ICDE, 1998.

[94] Y. Zhang, X. Lin, Y. Tao, and W. Zhang, *Uncertain location based range aggregates in a multi-dimensional space*, in Proceedings 25th IEEE International Conference on Data Engineering, 2009.

[95] Z. Zhang, K. Huang, and T. Tan, *Comparison of similarity measures for trajectory clustering in outdoor surveillance scenes*, in ICPR, 2006.

[96] Y. Zheng, H. Fu, X. Xie, W.-Y. Ma, and Q. Li, *Geolife GPS trajectory dataset - User Guide*, July 2011.