Algorithmic and Geometric Aspects of Data Depth With Focus on β-skeleton Depth

by

Rasoul Shahsavarifar

Master of Science, University of Tabriz, Iran, 2008
Bachelor of Science, Razi University, Iran, 2006

A DISSERTATION SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

Doctor of Philosophy

In the Graduate Academic Unit of Computer Science

Supervisor(s):	David Bremner, Ph.D, Computer Science
Examining Board:	Patricia Evans, Ph.D, Computer Science, Chair
	Suprio Ray, Ph.D, Computer Science
	Jeffrey Picka, Ph.D Mathematics and Statistics
External Examiner:	Pat Morin, Ph.D, Computer Science, Carleton University

This dissertation is accepted by

Dean of Graduate Studies

THE UNIVERSITY OF NEW BRUNSWICK

April, 2019

©Rasoul Shahsavarifar, 2019

Abstract

The statistical rank tests play important roles in univariate non-parametric data analysis. If one attempts to generalize the rank tests to a multivariate case, the problem of defining a multivariate order will occur. It is not clear how to define a multivariate order or statistical rank in a meaningful way. One approach to overcome this problem is to use the notion of data depth which measures the centrality of a point with respect to a given data set. In other words, a data depth can be applied to indicate how deep a point is located with respect to a given data set. Using data depth, a multivariate order can be defined by ordering the data points according to their depth values. Various notions of data depth have been introduced over the last decades. In this thesis, we discuss three depth functions: two well-known depth functions halfspace depth and simplicial depth, and one recently defined depth function named as β-skeleton depth, $\beta \geq 1$. The β-skeleton depth is equivalent to the previously defined spherical depth and lens depth when $\beta = 1$ and $\beta = 2$, respectively. Our main focus in this thesis is to explore the geometric and algorithmic aspects of β-skeleton depth. For the geometric part, we study

the arrangement of circles and lenses as the influence regions of β-skeleton depth. The combinatorial complexity of such arrangement is computed in this study. For the algorithmic part, we develop some algorithms to solve the problem of computing the planar β-skeleton depth and halfspace depth. The lower bound for the complexity of computation of planar β-skeleton depth is also investigated. Finally, we determine some relationships among different depth functions, and propose a method to approximate one depth function using another one. The obtained theoretical results are supported by some experimental results provided in the last chapter of this thesis.

Dedication

This thesis is dedicated to the memory of:

- my beloved sister, Masoumeh.
- my cousins, Mohsen and Alireza who were my best friends.

Acknowledgements

First of all, I wish to express my profound gratitude and appreciation to my supervisor, Dr. David Bremner, for his guidance, support and patience during my Ph.D studies. In the numberless meetings that we had together, he always took time to discuss our approaches and problems. I am always grateful to David for all that he did for me. I would also like to thank Diane Souvaine in Tufts University, Martin and Erik Demaine in MIT, and Reza Modarres in George Washington University for hosting me as a visiting student. Additionally, I thank people in CCCG2017 and CCCG2018 for useful discussions regarding the approximation of depth functions. Special thanks goes to all of my family members in Iran and Canada for always loving, encouraging, and supporting me. I am extremely fortunate for having such family. Finally, I would like to acknowledge my thesis committee members Dr. Patricia Evans, Dr. Huajie Zhang, Dr. Suprio Ray, Dr. Jeffrey Picka, and my external reviewer Dr. Pat Morin for reading this thesis carefully, and providing me with detailed and useful comments.

Table of Contents

Abstract **ii**

Dedication **iv**

Acknowledgements **v**

Table of Contents **ix**

List of Tables **xi**

List of Figures **xiii**

Abbreviations **xiv**

1 Introduction **1**

1.1 General Definition of Data Depth 2

1.2 Quick Review of HD, SD, and SkD_β 4

1.3 Overview of this Thesis 7

2 Background **10**

2.1 Computational Geometry Review 10

2.1.1 Model of Computation 10
2.1.2 Arrangements 12
2.1.3 Range Query 14
2.2 Non Computational Geometry Review 15
2.2.1 Poset 16
2.2.2 Fitting Function 17
2.2.3 Evaluation of Fitting Function 17
2.2.3.1 Coefficient of Determination 18
2.2.3.2 *AIC* and *BIC* 19

3 Properties of Data Depth **20**
3.1 General Framework 20
3.2 Halfspace Depth 24
3.3 Simplicial Depth 28
3.4 β-skeleton Depth 30
3.4.1 Spherical Depth and Lens Depth 33

4 Geometric Results in $\mathbb{R}^2$ **35**
4.1 Combinatorial Complexity 35
4.2 Geometric Properties of β-skeleton Depth 46

5 Algorithmic Results in $\mathbb{R}^2$ **50**
5.1 Optimal Algorithm to Compute Planar Spherical Depth 51
5.2 Algorithm to Compute Planar β-skeleton Depth when $\beta > 1$. 55

5.3 Algorithm to Compute Planar Halfspace Depth 61
5.4 Pseudocode 63

6 Lower Bounds **68**

6.1 Lower Bound for the Planar β-skeleton Depth, $\beta = 1$ 68
6.2 Lower Bound for the Planar β-skeleton Depth, $1 < \beta < \infty$ . . 72
6.3 Lower Bound for the Planar β-skeleton Depth, $\beta = \infty$ 78

7 Relationships and Experiments **82**

7.1 Geometric Relationships 83
7.1.1 Convergence of β-skeleton Depth 83
7.1.2 β-skeleton Depth versus Simplicial Depth 88
7.2 Relationships via Dissimilarity Measures 92
7.2.1 Fitting Function and Dissimilarity Measure 93
7.2.2 Dissimilarity Measure Between two Posets 94
7.3 Approximation of Halfspace Depth 95
7.3.1 Approximation of Halfspace Depth and Fitting Function 96
7.3.2 Approximation of Halfspace Depth and Poset Dissimilarity 96
7.4 Experimental Results 97
7.4.1 Experiments for Geometric Relationships 98
7.4.2 Experiments for Approximations 99

8 Conclusion **110**

8.1 Contributions 110
8.2 Open Problems and Directions for Future Work 112
8.2.1 Future Work 113

Bibliography **126**

Vita

List of Tables

4.1 The number of faces in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points. 38

4.2 The number of edges in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points. 39

4.3 The number of vertices in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points. 39

7.1 Summary of experiments for the geometric relationships among simplicial depth (SD), spherical depth ($SphD$), and lens depth (LD) 99

7.2 Summary of experiments for the convergence of β-skeleton depth 101

7.3 Summary of experiments for the halfspace depth approximation via a quadratic model 101

7.4 Summary of experiments for the halfspace depth approximation via a power model 101

List of Figures

2.1 The planar arrangement of 5 lines 13

2.2 Different posets and relations among their elements 16

3.1 D is invariant to affine transformation A 21

3.2 D vanishes at infinity 22

3.3 D is monotone on the rays 23

3.4 D is upper semi-continuous 23

3.5 Two examples of halfspace depth in the plane 25

3.6 halfspace depth 26

3.7 Two examples of simplicial depth in the plane 28

3.8 Simplicial depth contours 30

3.9 The β-influence regions defined by x_i and x_j for β=1, 2, 3, and ∞. 31

3.10 Partitions of the plane by β-skeleton depth with respect to $\{p_1, p_2, p_3\}$. 33

4.1 Arrangement of β-influence regions ($\beta = 1$) for collinear points 36

4.2 Visualization of rows $n = 2$ and $n = 3$ of Table 4.1, Table 4.2, and Table 4.3 38
4.3 Two arbitrarily crossed β-influence regions, $\beta = 3/2$. 44
4.4 Spherical depth of planar points with respect to $\{p_1, p_2, p_3, p_4\}$ 46
4.5 Lens depth of planar points with respect to $\{p_1, p_2, p_3, p_4\}$ . . . 47

5.1 Point t and spherical influence region $Sph(a, b)$ 53
5.2 $\theta(x_i) \in (\frac{\pi}{2}, \frac{3\pi}{2}]$ (left figure), and $\theta(x_i) \notin (\frac{\pi}{2}, \frac{3\pi}{2}]$ (right figure) . . 55
5.3 $H_o(p)$ and $B(c, r)$ defined by $a \in \mathbb{R}^2$ for $\beta = 1,\ 1.5,\ 2,$ and $\beta = \infty$, where $A = H_o(p) \setminus \{intB_o(c, r)\}$ 56

6.1 An illustration of A and S for $\beta = 1$ 71
6.2 An illustration of B and S for $\beta = 2$ 74
6.3 The β-influence region for $a, b \in \mathbb{R}^2$. 76
6.4 The β-influence regions for some elements of S when $1 < \beta < \infty$. 78
6.5 An illustration of B and S when $\beta = \infty$ 79

7.1 $S_\beta(a, b)$ and $S_{\beta'}(a, b)$ 85
7.2 Slab and β_{ij}-influence region containing point q 86
7.3 Triangle abc and its corresponding spherical influence regions containing q 90
7.4 $Sph(x_i, x_j)$ and all triangles, with edge $x_j x_j$, containing q . . . 91
7.5 Fitting a linear model to SkD_∞ and SkD_1. 102
7.6 Fitting a linear model to SkD_∞ and SkD_2 102
7.7 Fitting a linear model to SkD_∞ and SkD_3. 103

7.8 Fitting a linear model to SkD_{∞} and SkD_{1000}. 103
7.9 Fitting a linear model to SkD_{∞} and SkD_{10000}. 104
7.10 Fitting a quadratic model to HD and SkD_1. 104
7.11 Fitting a quadratic model to HD and SkD_2. 105
7.12 Fitting a quadratic model to HD and SkD_3. 105
7.13 Fitting a quadratic model to HD and SkD_{1000}. 106
7.14 Fitting a quadratic model to HD and SkD_{10000}. 106
7.15 Fitting a power model to HD versus SkD_1. 107
7.16 Fitting a power model to HD and SkD_2. 107
7.17 Fitting a power model to HD and SkD_3. 108
7.18 Fitting a power model to HD and SkD_{1000}. 108
7.19 Fitting a power model to HD and SkD_{10000}. 109

List of Symbols, Nomenclature or Abbreviations

D	a data depth function
$D(q; S)$	depth value of point q with respect to data set S
$Conv$	convex hull
I	indicator function
inf	infimum
int	interior
H	halfspace
$\mathbb{H}$	class of closed halfspaces
HD	halfspace depth
SD	simplicial depth
SkD_β	β-skeleton depth
$SphD$	spherical depth
LD	lens depth
$S_\beta(x_i, x_j)$	β-influence region corresponding to points x_i and x_j
$Sph(x_i, x_j)$	spherical influence region corresponding to points x_i and x_j
$L(x_i, x_j)$	lens influence region corresponding to points x_i and x_j
CC	combinatorial complexity
GG	Gabriel graph
RNG	relative neighbourhood graph

Chapter 1

Introduction

For a univariate data set of n points with unknown distribution, consider the problem of computing a point (location estimator) which best summarizes the data set. One may answer this problem by introducing the mean of data set as the desired point. This answer is acceptable as the points are some minimum relative distance apart. However, in general, it is not the best choice because it is enough to move one of the data points very far away from the rest. This causes the mean to follow such single point, but not the majority of the data points. From this example, it can be deduced that the high level of robustness is an important characteristic that a location estimator should have [18]. This characteristic of an estimator can be measured by a factor called the *breakdown point*, which is the portion of data points that can move to infinity before the estimator does the same [59, 35]. By this definition, the breakdown point of the mean is $1/n$, whereas the breakdown

point of the median is 1/2. It is proven that the maximum breakdown point for any location estimator is 1/2 [60]. As such, median is an appropriate location estimator for an ordered univariate data set [17].

If one attempts to generalize the concept of median to higher dimensions, an additional problem will occur. In this case, it is not clear how to define a multivariate order that can be applied to compute the median of data set. One approach is to use the notion of data depth which we study in this thesis.

1.1 General Definition of Data Depth

Generally speaking, a data depth is a measure in non-parametric multivariate data analysis that indicates how deep (central) a point is located with respect to a given data set in $\mathbb{R}^d$ ($d \in \mathbb{Z}^+$). In other words, data depth introduces a partial order relation on $\mathbb{R}^d$ because it assigns a corresponding rank (the depth of point with respect to a given data set) to every point in $\mathbb{R}^d$. As a result, applying a data depth on a data set generates a partial ordered set (*poset*)[1] of the data points. Considering different depth values, each data depth determines a family of *regions*. Each region contains all points in $\mathbb{R}^d$ with the same depth values. Among all regions, the *central* region also known as the *median set* is the one whose depth is maximum. Inside the central region a point in $\mathbb{R}^d$ (not necessarily from the data set) with the largest depth

[1] A poset is a set together with a partial ordering relation which is reflexive, antisymmetric and transitive.

is called the *median* of the data set.

As discussed above, defining a multivariate order and generalizing the concept of median are not straightforward. Over the last few decades, various notions of data depth have been introduced as powerful tools in non-parametric multivariate data analysis. Most of them have been defined to solve specific problems. A short list of the most important and well-known depth functions is as follows: *halfspace depth* [48, 81, 86], *simplicial depth* [54], *Oja depth* [68], *regression depth* [72], *majority depth* [57], *Mahalanobis depth* [61], *projection depth* [90], *Zonoid depth* [37], *spherical depth* [38]. These depth functions are different in application, definition, and geometry of their central regions. Each data depth has different properties and requires different time complexity to compute. In 2000, Zuo and Serfling [90] provided a general framework for statistical depth functions. In this framework, a data depth is a real valued function that possesses the following properties: *affine invariance*, *maximality at the center*, *monotonicity on rays*, and *vanishing at infinity*. However depending on the particular application, not every depth function needs to fit in this framework. For example, in a medical study that includes patient data such as height and weight perhaps the affine invariance is not an important requirement because the height and weight axes are meaningful. In fact, we need the data depth used in this medical study to be invariant under scaling of the axes. Given this property, the results would be independent from the measuring systems.

The concept of data depth is widely studied by statisticians and computational geometers. Some directions that have been considered by researchers include defining new depth functions, developing new algorithms, improving the complexity of computations, computing both exact and approximate depth values, and computing depth functions in lower and higher dimensions. Two surveys by Aloupis [11] and Small [81] can be referred as overviews of data depth from a computational geometer's and a statistician's point of view, respectively. Some research on algorithmic aspects of planar depth functions can be found in [10, 13, 21, 25, 26, 30, 56, 62, 72].

The main focus of this thesis is on the geometric and algorithmic concepts of three depth functions: halfspace depth (HD), simplicial depth (SD), and β-skeleton depth (SkD_β), briefly reviewed in Section 1.2. More detailed definitions and properties of these depth functions are explored in Chapter 3. The β-skeleton depth is not as well-known as the other depth functions. However, it is easy to compute even in higher dimensions.

1.2 Quick Review of HD, SD, and SkD_β

In 1975, Tukey generalized the definition of univariate median and defined the halfspace median as a point in which the halfspace depth is maximized, where the halfspace depth is a multivariate measure of centrality of data

points. Halfspace depth is also known as Tukey depth or location depth. In general, the halfspace depth of a query point q with respect to a given data set S is the smallest fraction of data points that are contained in a closed halfspace through q [16, 21, 78, 86]. The halfspace depth function has various properties such as vanishing at infinity, affine invariance, and decreasing along rays. These properties are proved in [34]. Many different algorithms for the computation of halfspace depth in lower dimensions have been developed [21, 22, 25, 75]. The bivariate and trivariate case of halfspace depth can be computed exactly in $O(n \log n)$ and $O(n^2 \log n)$ time [73, 83], respectively. However, computing the halfspace depth of a query point with respect to a data set of size n in dimension d is an NP-hard problem if both n and d are part of the input [50]. Due to the hardness of the problem, designing efficient algorithms to compute and approximate the halfspace depth of a point remains an interesting task in the research area of data depth [2, 15, 27, 45].

In 1990, another generalization of univariate median was presented by Liu [54]. This generalization is based on this fact that the median of a data set $S \subseteq \mathbb{R}$ is a point in $\mathbb{R}$ which is contained in the maximum number of closed intervals, formed by any pair of data points. Liu replaced the closed intervals by closed simplices[2] in higher dimensions and presented the definition of *simplicial median*. In some references (e.g. [55]) open simplices are considered. In this thesis, we only consider closed simplices as originally used

[2]A simplex in $\mathbb{R}^1$ is a line segment, in $\mathbb{R}^2$ is a triangle, in $\mathbb{R}^3$ is a tetrahedron, etc.

by Liu in the definition of simplicial median. For a data set $S \in \mathbb{R}^d$, the simplicial median set is a set of points in $\mathbb{R}^d$ which are contained in the maximum number of closed simplices formed by any $(d+1)$ data points from S. The definition of simplicial median provides another measure of centrality of point $q \in \mathbb{R}^d$ with respect to the data set $S \subseteq \mathbb{R}^d$. This measure is known as simplicial depth which is the proportion of all closed simplices obtained from S that contain q. The straightforward algorithm to compute the simplicial depth takes $O(n^{d+1})$ time. The simplicial depth is affine invariant [54]. It is proved that the breakdown point of the simplicial median is worse than the breakdown point of the halfspace median [28]. For a set of n points in general position[3] in $\mathbb{R}^2$, the depth of the simplicial median multiplied by $\binom{n}{3}$ is $\Theta(n^3)$ [9]. The bivariate case of simplicial depth can be computed optimally in $\Theta(n \log n)$ time [73, 12].

In 2006, Elmore, Hettmansperger, and Xuan [38] defined another notion of data depth named *spherical depth*. It is defined as the proportion of all closed hyperballs with the diameter $\overline{x_i x_j}$, where x_i and x_j are any pair of points in the given data set S. These closed hyperballs are known as influence regions of the spherical depth function. In 2011, Liu and Modarres [58], modified the definition of influence region, and defined *lens depth*. Each lens depth influence region is defined as the intersection of two closed hyperballs $B(x_i, \|x_i, x_j\|)$ and $B(x_j, \|x_i, x_j\|)$. These influence regions of spher-

[3]A data set in $\mathbb{R}^2$ is in general position if no three points are collinear.

ical depth and lens depth are the multidimensional generalization of *Gabriel circles* and lunes in the definition of the *Gabriel Graph* [41] and *Relative Neighbourhood Graph* [84], respectively. In 2017, Yang [89], generalized the definition of influence region, and introduced a familly of depth functions called β-skeleton depth, indexed by a single parameter $\beta \geq 1$. The influence region of β-skeleton depth is defined to be the intersection of two closed hyperballs given by $B(c_i, \frac{\beta}{2}\|x_i, x_j\|)$ and $B(c_j, \frac{\beta}{2}\|x_i, x_j\|)$, where c_i and c_j are some combinations of x_i and x_j. Spherical depth and lens depth can be obtained from β-skeleton depth by considering $\beta = 1$ and $\beta = 2$, respectively. The β-skeleton depth has some nice properties including symmetry about the center, maximality at the centre, vanishing at infinity, and monotonicity. Depending on whether Euclidean distance or Mahalanobis distance is used to construct the influence regions, the β-skeleton depth can be orthogonally invariant or affinely invariant. All of these properties are explored in [38, 58, 88, 89]. A notable characteristic of the β-skeleton depth is that its time complexity is $O(dn^2)$ which grows linearly in the dimension d.

1.3 Overview of this Thesis

In this chapter we have introduced the notion of data depth, and reviewed the definitions of the halfspace depth, simplicial depth, and β-skeleton depth. In Chapter 2 we recall some concepts in computational geometry and statistics which are used in this thesis. Chapter 3 includes a general framework

for depth functions. The properties and the previous results related to the halfspace depth, simplicial depth, and β-skeleton depth are also explored in this chapter. The main contributions of this thesis are provided in the following chapters.

In Chapter 4, we study the β-skeleton depth from a geometric perspectives. We obtain an exact bound $O(n^4)$ for the combinatorial complexity of the arrangement[4] of planar β-influence regions.

In chapter 5, we present an optimal algorithm to compute the planar spherical depth (i.e. β-skeleton depth, $\beta = 1$) of a query point. This algorithm takes $\Theta(n \log n)$ time. In Theorem 5.2.1, we prove that computing the planar β-skelton depth of a query point can be reduced to a combination of at most $3n$ range counting problems. This reduction and the results in [4, 14, 44, 80], allow us to develop an algorithm for computing the planar β-skeleton depth. This algorithm takes $O(n^{(3/2)+\epsilon})$ query time. In the last section of this chapter, we present an $\Theta(n \log n)$ algorithm for computing the planar halfspace depth. Although there exist another algorithm with the same running time for this problem [9], our algorithm is much simpler in terms of implementation. We use the specialized halfspace range query that is explored in Algorithm 5.

Chapter 6 includes a proof of the lower bound for computing the planar β-skeleton depth for cases $\beta = 1, 1 < \beta < \infty$, and $\beta = \infty$, separately. In each

[4]An arrangement of some geometric objects is a subdivision of space formed by a collection of such objects.

case, we reduce the problem of Element Uniqueness[5] which takes $\Omega(n \log n)$ to the problem of computing the planar β-skeleton depth of a query point.

In Chapter 7, we study the relationships among different depth notions in this thesis. These relationships are explored in two different ways. First, we use the geometric properties of the influence regions, and prove that the simplicial depth has an upper bound in terms of a constant factor of β-skeleton depth (i.e. $SD \leq 3/2\, SphD$). Secondly, for every pair of depth functions, we employ a fitting function to approximate one data depth using another one. For example, considering a certain amount of error, we approximate the planar halfspace depth by a polynomial function of planar β-skeleton depth, $\beta \to \infty$. In the last Section 7.4, some experimental results are provided to support the the theoretical results in this chapter.

Finally, Chapter 8 is the conclusion of this thesis.

[5]Element Uniqueness problem: Given a set $A = \{a_1, ..., a_n\}$ of real numbers, is there a pair of indices i, j with $i \neq j$ such that $a_i = a_j$?

Chapter 2

Background

In this chapter, we present some background for the later chapters of this thesis. The background is provided in two sections: computational geometry review (Section 2.1) and non-computational geometry review (Section 2.2).

2.1 Computational Geometry Review

Some concepts in computational geometry which are frequently referred to in this thesis are briefly reviewed in this section.

2.1.1 Model of Computation

One of the fundamental steps in any study of algorithms is to specify the adopted model of computation. In particular, it is not possible to determine

the complexity of an algorithm without knowing the primitive operations[1] employed in such algorithm. Choosing an appropriate model of computation is closely related to the nature of given problem. The problems in computational geometry are of different types. However, they may be generally categorized as: *Subset Selection*, *Computation*, and *Decision*. Naturally, for each of subset selection and computation problems, there is a decision problem. As an example, a YES/NO answer is needed for the question: "For a given set S, does the the set $S' \subset S$ satisfies a certain property P?". It can be discussed that we need to deal with the real numbers to answer some problems in the above categories. Suppose, for example, that in a given set of points with integer coordinates, the distance between a pair is requested. As such, we need a *random-access machine* in which each storage location is capable to hold a single real number. The related model of computation is known as the *real* ***RAM*** [6, 69]. The primitive operations with unit cost in this model are as follows:

- arithmetic operations ($+,-,\times,\div$).
- comparison between two real numbers ($<,\leq,=,\neq,\geq,>$).
- indirect addressing of memory
- k-th root, trigonometric functions, exponential functions, and logarithmic functions.

[1]The primitive operations are those whose costs are constant for the fixed-length operands.

The execution of a decision making algorithm in the real **RAM** can be described as a sequence of operations of two types: arithmetic and comparisons. Depending on the outcome of the arithmetic, the algorithm has a branching option at each comparison. Hence, the computation which is executed by the real **RAM** can be considered as a path in a rooted tree. In the literature, this rooted tree is known as the *algebraic decision tree* model of computation [32, 70, 71, 82]. An algebraic decision tree on a set of variables $\{x_1, ..., x_k\}$ is a program with a finite sequence of statements. Each of these statements is obtained based on the result of $f(x_1, ..., x_k) \diamond 0$, where $f(x_1, ..., x_k)$ is an algebraic function and $\diamond$ denotes any comparison relation. For a set of n real elements, Ben-Or proved that any algebraic computation tree that solves the element uniqueness problem must have the complexity of at least $\Omega(n \log n)$ [19].

2.1.2 Arrangements

Suppose that $\Gamma = \{\gamma_1, ..., \gamma_n\}$ is a finite collection of geometric objects in $\mathbb{R}^d$. The arrangement $\mathcal{A}(\Gamma)$ is the subdivision of $\mathbb{R}^d$ into connected cells of dimensions $0, 1, ..., d$ induced by Γ, where each k-dimensional cell is a maximal connected subset of $\mathbb{R}^d$ determined by a fixed number of $\gamma_i \in \Gamma$. The arrangement $\mathcal{A}(\Gamma)$ is planar if $\gamma_i \subset \mathbb{R}^2$. In the planar arrangement $\mathcal{A}(\Gamma)$, a 0-dimensional cell is a *vertex*, a 1-dimensional cell is an *edge*, and a 2-dimensional cell is a *face*. As an illustration, Figure 2.1 represents the arrangement of 5 given lines in the plane. In studying arrangements, one

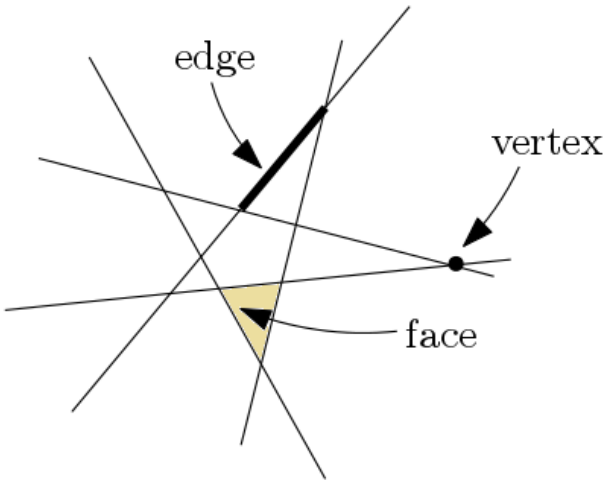

Figure 2.1: The planar arrangement of 5 lines

of the fundamental questions that needs to be answered is how complex each arrangement can be. Computing the combinatorial complexity (CC) of each arrangement helps to answer this question, where the CC of an arrangement $\mathcal{A}(\Gamma)$ is the total number of cells of $\mathcal{A}(\Gamma)$. For example, in Figure 2.1, $CC = 42$ because there are 25 edges, 10 vertices, and 7 faces in the arrangement. Every planar arrangement $\mathcal{A}(\Gamma)$ can be presented by a planar graph $G(V, E)$ such that $v_i \in V$ if and only if face $f_i \in \mathcal{A}(\Gamma)$. Furthermore, $e_{ij} \in E$ if and only if f_i and f_j are two adjacent faces of $\mathcal{A}(\Gamma)$. We recall Euler's Formula [40, 49] which is one of the most useful facts regarding a planar graph.

Theorem 2.1.1. Euler's Formula *If G is a planar graph with C_v vertices, C_e edges, and C_f faces,*

$$C_v - C_e + C_f = 2.$$

Because we study depth functions in this thesis, we define a nonstandard terminology that is frequently referred to in Chapter 4.

Definition 2.1.1. *In a planar arrangement $\mathcal{A}(\Gamma)$, we define a depth region to be the maximal connected union of faces with the same depth value.*

Further and more detailed results on the arrangements of hyperplanes, spheres, simplices, line segments can be found in [5, 20, 43].

2.1.3 Range Query

Range query is among the central problems in computational geometry. In fact, many problems in computational geometry can be represented as a range query problem. In this thesis, computing the depth value of a query point in some cases is converted to a range query problem (see Chapter 5). In a typical range query problem, we are given a data set S of n points, and a family Λ of *ranges* (i.e. subsets of $\mathbb{R}^d$). S should be preprocessed into a data structure such that for a query range $\tau \in \Lambda$, all points in $S \cap \tau$ can be efficiently reported or counted. In addition to *range counting* and *range reporting*, another type of range query problems is *range emptiness* (i.e. checking if $S \cap \tau = \emptyset$). Due to the nature of related problems in data depth among the three types of range query problems, range reporting is not used in this thesis. Typical example of ranges in range query problems include halfspaces, simplices, rectangle. However, in many applications we may need to deal with ranges bounded by nonlinear functions. In other words,

the ranges are some *semialgebraic sets*[2]. In this case, the problem is known as semialgebraic range query [3, 4, 43, 80]. The recent results on some range query problems are reviewed in Chapter 5.

In some problems such as halfspace and simplex range query, solving the exact range counting is expensive which means that no exact algorithm with both near linear storage complexity and preprocessing time, and low query time exists [1]. This issue is a motivation for seeking some approximation techniques that can be applied to approximate a range counting problem. One way to achieve such technique is through the notion of (ε, τ)-approximation, which is defined in the following.

Definition 2.1.2. *Given a set $S = \{x_1, ..., x_n\} \subset \mathbb{R}^2$, a parameter $\varepsilon > 0$, and a semialgebraic range τ with constant description complexity, we say that algorithm A computes an (ε, τ)-approximation η_τ if*

$$(1 - \varepsilon)|S \cap \tau| \leq \eta_\tau \leq (1 + \varepsilon)|S \cap \tau|.$$

2.2 Non Computational Geometry Review

In this section, we provide some background regarding the concepts of partially ordered set (poset), fitting function, and the measures of goodness of statistical models. These concepts are used in Sections 7.2 and 7.3, where we approximate depth functions using the idea of a fitting function.

[2]A semialgebraic set is a subset of $\mathbb{R}^d$ obtained from a finite Boolean combination of d-variate polynomial equations and inequalities.

2.2.1 Poset

For a finite set $S = \{x_1, ..., x_n\}$, it is said that $P = (S, \preceq)$ is a poset if $\preceq$ is a partial order relation on S, that is, for all $x_i, x_j, x_t \in S$:

i) $x_i \preceq x_i$.

ii) $x_i \preceq x_j$ and $x_j \preceq x_t$ implies that $x_i \preceq x_t$.

iii) $x_i \preceq x_j$ and $x_j \preceq x_i$ implies that $x_i \equiv_p x_j$, where $\equiv_p$ is the corresponding equivalence relation.

Poset $P = (S, \preceq)$ is called a chain if any two elements of S are comparable, i.e., given $x_i, x_j \in S$, either $x_i \preceq x_j$ or $x_j \preceq x_i$. If there is no comparable pair among the elements of S, the corresponding poset is an anti chain. Figure 2.2 illustrates different posets with the same elements.

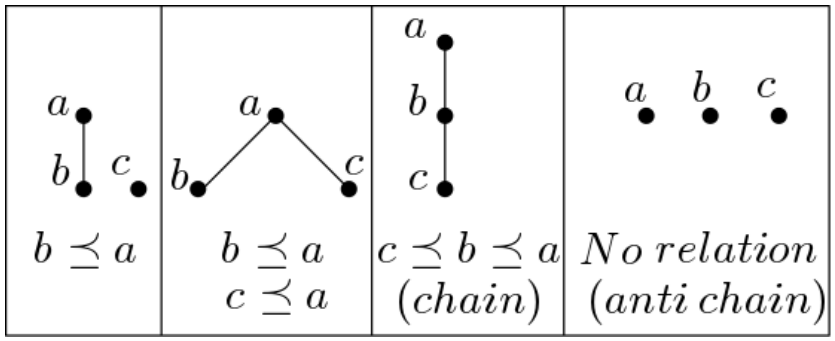

Figure 2.2: Different posets and relations among their elements

2.2.2 Fitting Function

One method to represent a set of data is to use the notion of fitting function. This representation technique involves choosing values for the parameters in a function to best describe the data set. Depending on whether the number of parameters is known or unknown in advance, the method is called *parametric fitting* or *nonparametric fitting*, respectively [42]. In *depth function approximation* that we study in this thesis, the number of parameters is known; therefore, we focus on the parametric fitting. Consider a set $\omega = \{(v_i, u_i); 1 \leq i \leq n\}$, where $V = (v_1, ..., v_n)$ and $U = (u_1, ..., u_n)$ are two given vectors. The goal is to obtain a function f that best describe ω (i.e. f captures the trend among the elements of ω). For every element in ω, $u_i = f(v_i) \pm \delta_i$, where $\delta_i \geq 0$ is the error corresponding to the approximation of u_i.

Obviously, there may be more than one candidate for f. The question is how to measure the goodness of fit in order to obtain the best fitting function.

2.2.3 Evaluation of Fitting Function

In this section we review some methods and criteria that can be employed to measure how good a fitting function is. Based on the values of these methods, we can select the best fitting function for our data points.

2.2.3.1 Coefficient of Determination

For two vectors $U = (u_1, ..., u_n)$ and $V = (v_1, ..., v_n)$, suppose that f is a function such that $u_i = f(v_i) \pm \delta_i$. Let $\xi_i = u_i - \overline{U}$, where $\overline{U}$ is the average of all $u_i; 1 \leq i \leq n$. The coefficient of determination [77] of f is indicated by R^2 (or r^2), and defined by:

$$r^2 = \frac{\sum_{i=1}^{n}(\xi_i^2 - \delta_i^2)}{\sum_{i=1}^{n}\xi_i^2}. \tag{2.1}$$

To avoid confusion between $\mathbb{R}^2$ and R^2, hereafter in this thesis, the coefficient of determination is represented by r^2 alone. In statistics and data analysis, r^2 is applied as a measure that assesses the ability of a model in predicting the real value of a parameter. From Equation (2.1), it can be deduced that the value of r^2 is always within the interval $[0, 1]$. Generally, a high value of r^2 related to a model fitted to a set of data indicates that the model is a good fit for such data. Obviously, the interpretations of fit depend on the context of analysis. For example, a model with $r^2 = 0.9$ explains 90% of the variability of the response data around its mean. This percentage may be considered as a high value in a social study. However, it is not a high enough value in some other research areas (e.g. biochemistry, chemistry, physics), where the value of r^2 could be much closer to 100 percent.

2.2.3.2 *AIC* and *BIC*

In statistics, *Akaike Information Criterion (AIC)* [7] and the *Bayesian information criterion (BIC)* [76] are two commonly used criteria to select the best model among a collection of candidates that fit to a given data set. These two criteria provide a standardized way to achieve a balance between the goodness of fit and the simplicity of the model. The model with the lowest value of AIC (and/or the lowest value of BIC) is preferred. Among a finite number of candidate models M_k fitted to a data set S, both AIC and BIC involve choosing the model with the best penalized log-likelihood. The likelihood function corresponding to each model M_k is a conditional probability given by $p(S|P_k)$, where P_k is the parameter vector of M_k. In the literature, it is more common to use the log-likelihood $\mathfrak{L}(P_k) = \log p(S|P_k)$ [67]. Considering the above notations, the AIC and BIC values of every model can be computed using the following equations.

$$AIC = -2\mathfrak{L}(P_k) + 2|P_k| \tag{2.2}$$

$$BIC = -2\mathfrak{L}(P_k) + |P_k|\log(n), \tag{2.3}$$

where $|P_k|$ represents the number of parameters of M_k and n is the size of data set S. From Equations (2.2) and (2.3), it can be seen that BIC penalizes the model complexity more heavily. In particular, BIC prefers the models with less parameters as the size of S increases, whereas AIC does not penalize the model based on the size of S.

Chapter 3

Properties of Data Depth

In this chapter we review general properties of a depth function. We also discuss different notions of data depth such as halfspace depth, simplicial depth, and β-skeleton depth that are studied in this thesis.

3.1 General Framework

A data depth D, $(D : (\mathbb{R}^d, \mathbb{R}^{n\times d}) \to \mathbb{R}^+)$ is a real-valued function defined at any arbitrary point $q \in \mathbb{R}^d$ with respect to a given data set $S \subset \mathbb{R}^d$. A typical depth function D satisfies the following conditions which are known as the general framework for data depth, introduced in [90],.

- **Affine invariance** [34, 85]: For invertible matrix A and constant vector b, D is affinely invariant if

$$D(Aq + b; AS + b) = D(q; S); A \in \mathbb{R}^{d \times d}, b \in \mathbb{R}^d.$$

See Figure 3.1. If this equation holds for any orthogonal matrix A (i.e.$A^T A = AA^T = I$), D is *orthogonally invariant* which is weaker than affine invariant. Transforming data points is commonly used in processing a data set; therefore, it is important for a data depth to be invariant in some sense.

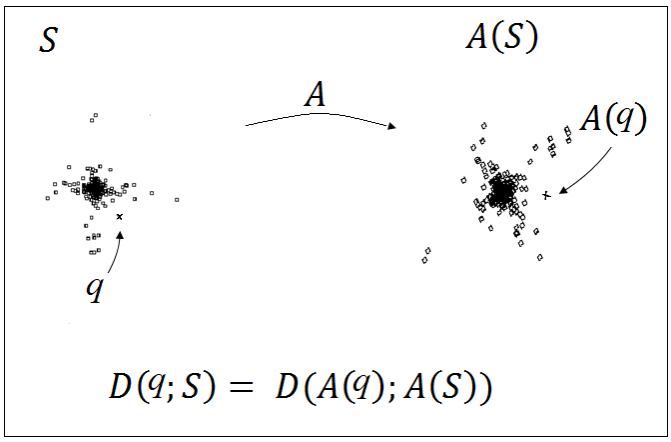

Figure 3.1: D is invariant to affine transformation A

- **Vanishing at infinity** [54, 90]: D vanishes at infinity if for every sequence $\{q_k\}_{k \in \mathbb{N}}$ with $\lim_{k \to \infty} ||q_k|| = \infty$,

$$\lim_{k \to \infty} D(q_k; S) = 0.$$

Figure 3.2 is a visualization of this property.

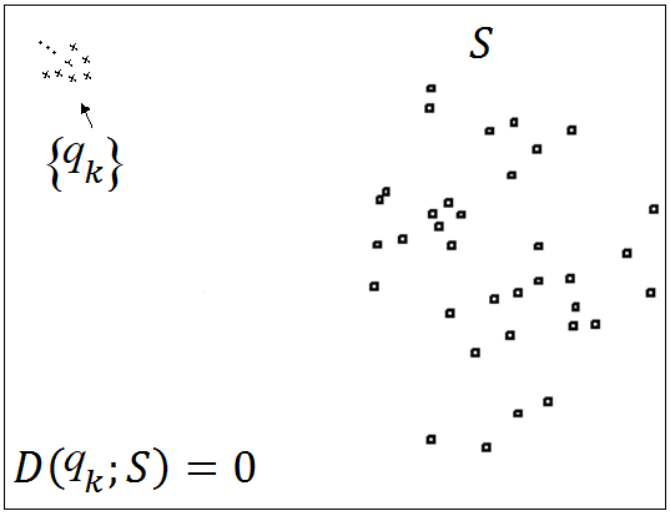

Figure 3.2: D vanishes at infinity

- **Monotone on rays** [17, 60, 57]: For $t \in \mathbb{R}^d$ as a center point (with maximal depth value), D is monotone on the rays if

$$D(q; S) \leq D(p; S) = D(t + r(q - t); S); r \in [0, 1],$$

where the point p is a convex combination of q and t. As an illustration, see Figure 3.3.

- **Upper semi-continuity** [36, 65]: For $\alpha \in \mathbb{R}^+$, D is upper semi-continuous if $D_\alpha(S)$ is a closed set, where

$$D_\alpha(S) = \{q \in \mathbb{R}^d | D(q; S) \geq \alpha\}.$$

The outer boundary of D_α is called the *contour* of depth α [34]. See

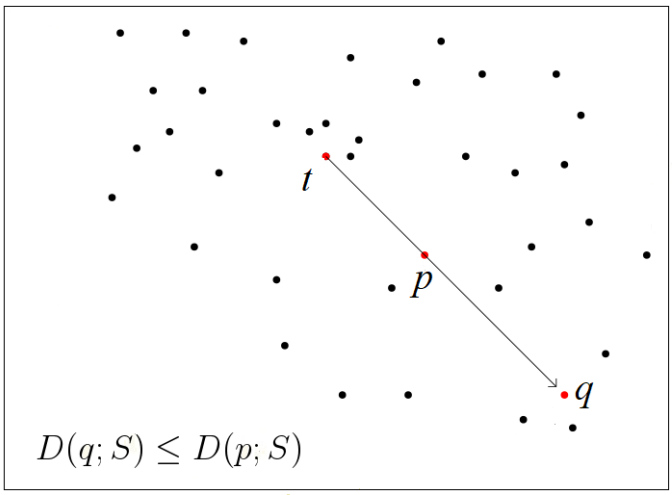

Figure 3.3: D is monotone on the rays

Figure 3.4.

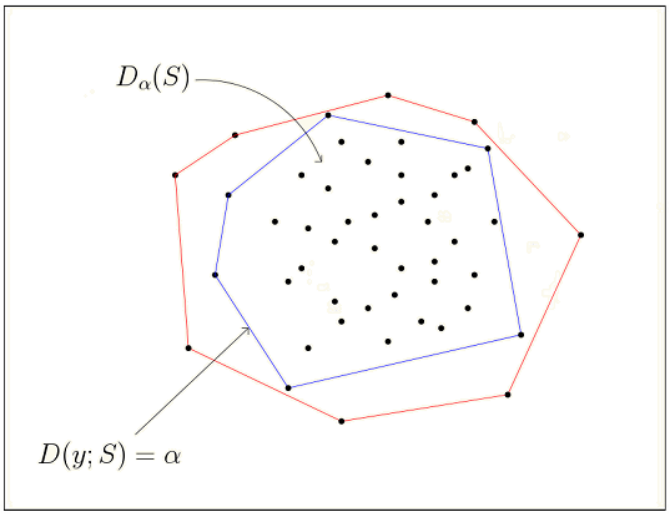

Figure 3.4: D is upper semi-continuous

In addition to these four properties, some other conditions such as high breakdown point and level-convex are discussed for depth functions [34, 35,

60].

- **High breakdown point** [35, 60]: The breakdown point of a location estimator $c = c(S)$ (i.e. the center point of $S = \{x_1, ..., x_n\}$ in $\mathbb{R}^d$) is a number between zero and one, introducing the proportion of data points that must be moved to infinity before c moves to infinity. In other words, the breakdown point $b(c)$ can be defined by

$$b(c) = \min_m \left\{ \frac{m}{n}; \sup_{S'} \|c(S) - c(S')\| = \infty \right\},$$

  where $S' = (S \setminus S_m) \cup Y$, $S_m \subseteq S$, $|S_m| = m$, and $Y = \{y_1, ..., y_m\}$ is an arbitrary set in $\mathbb{R}^d$.

- **Level-convex** [34]: Data depth D is level-convex if all of its corresponding contours D_α are convex.

3.2 Halfspace Depth

The halfspace depth of a query point $q \in \mathbb{R}^d$ with respect to a given data set $S = \{x_1, ..., x_n\} \subseteq \mathbb{R}^d$ is defined as the minimum portion of points of S contained in any closed halfspace that has q on its boundary. Using the notation of $HD(q; S)$, the above definition can be presented by (3.1).

$$HD(q; S) = \frac{2}{n} \min\{|S \cap H| : H \in \mathbb{H}\}, \tag{3.1}$$

where $2/n$ is the normalization factor[1], $\mathbb{H}$ is the class of all closed halfspaces in $\mathbb{R}^d$ that pass through q, and $|S \cap H|$ denotes the number of points within the intersection of S and H. As illustrated in Figure 3.5, $HD(q_1; S) = 6/13$ and $HD(q_2; S) = 0$, where S is a given set of points in the plane and q_1, q_2 are two query points not in S.

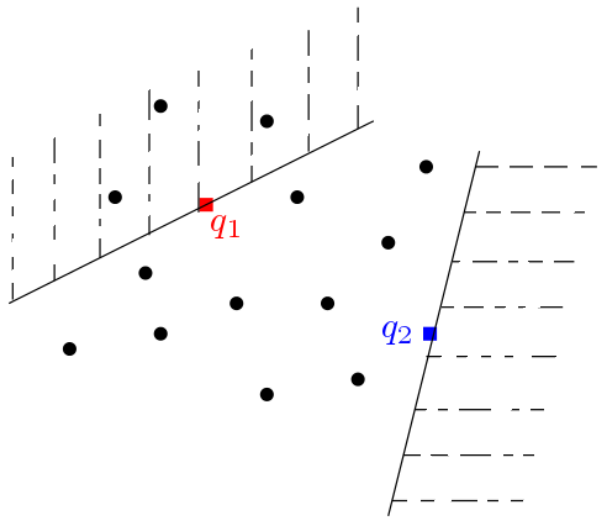

Figure 3.5: Two examples of halfspace depth in the plane

For a given data set S and a query point q, it can be verified that $HD(q; S)$ is equal to zero if and only if q is outside the convex hull[2] of S (see Figure 3.6). The halfspace depth satisfies all desirable properties of depth functions presented in Section 3.1 [86, 34]. Furthermore, for a data set in general position[3]

[1]Instead of the normalization factor $1/n$ which is common in literature, we use $2/n$ in order to let the halfspace depth of 1 to be achievable.

[2]The convex hull of a data set $S \subset \mathbb{R}^d$ is the smallest convex set in $\mathbb{R}^d$ that contains S.

[3]It is said that a data set, in $\mathbb{R}^d$, is in general position if no $(d+1)$ points of the data points lie on a common hyperplane.

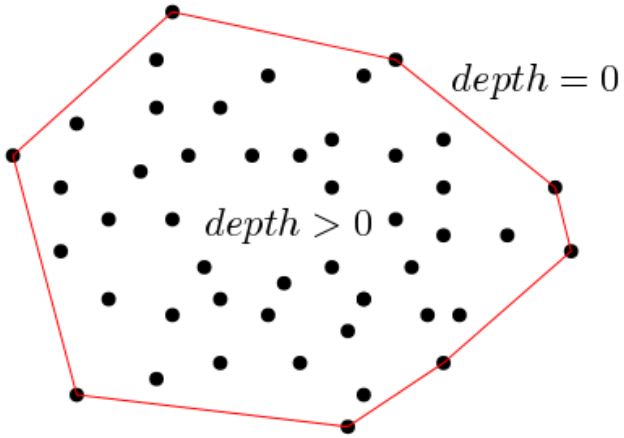

Figure 3.6: halfspace depth

in $\mathbb{R}^d$, the breakdown point of the halfspace median is at least $\frac{1}{d+1}$, and at most $1/3$ for $d \geq 2$ [34, 33]. Another nice property of the halfspace depth is that the depth contours are all convex and nested, i.e. the contour of unnormalized depth $(k+1)$ is convex, and geometrically surrounded by the contour of unnormalized depth k which is also convex [34].

Among all data depths, halfspace depth, perhaps, is the most extensively studied data depth. Many algorithms to compute, or approximate, the halfspace median have been developed in recent years [2, 22, 25, 31, 52, 62, 75, 87]. A summary of these results is as follows.

- A complicated $O(n^2 \log n)$ algorithm to compute the halfspace depth of a point in $\mathbb{R}^2$ is implemented by Rousseeuw and Ruts in [74].

- An optimal randomized algorithm for computing the halfspace median is developed by Chan [25]. This algorithm requires $O(n \log n)$ expected

time for n non-degenerate data points.

- An $O(n \log^5 n)$ algorithm to compute the bivariate halfspace median is presented by Matousek in [62]. The algorithm consists of two main steps: an $O(n \log^4 n)$ algorithm to compute any point with depth of at least k, and a binary search on k to find the median.

- By improving the Matousek's algorithm, Langerman provided an algorithm which computes the bivariate halfspace median in $O(n \log^3 n)$ [53].

- For a set on n non-degenerate points in $\mathbb{R}^d$, the halfspace median can be computed in $O(n \log n + n^{d-1})$ expected time (Theorem 3.2 in [25]).

- The halfspace depth of $q \in \mathbb{R}^d$ with respect to $S \subseteq \mathbb{R}^d$ can be computed in $O((d+1)^k LP(n, d-1))$ time, where k is the value of the output and $LP(n, d)$ denotes the running time for solving a linear program with n constraints and d variables (Theorem 5 in [21]).

- In a worst case scenario in $\mathbb{R}^2$, when the data points all are on the convex hull, it takes at least $O(n^2)$ to find all of the halfspace depth contours [63].

- A center point of $S \subseteq \mathbb{R}^d$, a point whose halfspace depth is at least $[\frac{n}{d+1}]/n$, can be computed in $O(n)$ [23].

- Computing the halfspace of a query point in $\mathbb{R}^2$ can be done in $\Theta(n \log n)$ time [9, 12].

3.3 Simplicial Depth

The simplicial depth of a query point $q \in \mathbb{R}^d$ with respect to $S = \{x_1, ..., x_n\} \subset \mathbb{R}^d$ is defined as the total number of the closed simplices formed by data points that contain q. This definition can be given by (3.2).

$$SD(q; S) = \frac{1}{\binom{n}{d+1}} \sum\nolimits_{(x_1,...,x_{d+1}) \in \binom{n}{d+1}} I(q \in Conv[x_1, ..., x_{d+1}]), \qquad (3.2)$$

where $1/\binom{n}{d+1}$ is the normalization factor, the convex hull $Conv[x_1, ..., x_{d+1}]$ is a closed simplex formed by $d+1$ points of S, and I is the indicator function. For $S = \{a, b, c, d, e\}$ in $\mathbb{R}^2$, Figure 3.7 illustrates that $SD(q_1; S) = 3/10$ and $SD(q_2; S) = 4/10$.

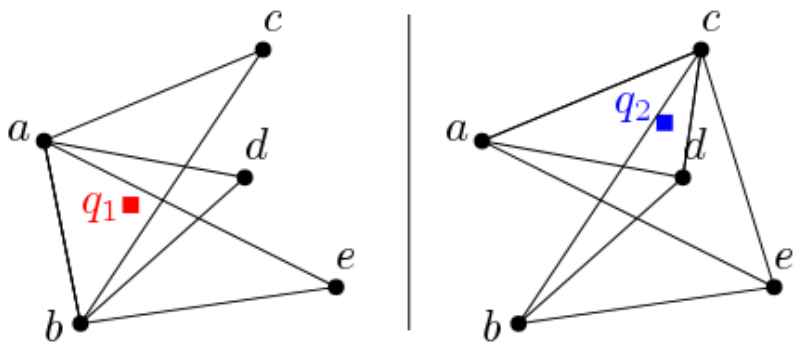

Figure 3.7: Two examples of simplicial depth in the plane

Liu [55] proved that the simplicial depth satisfies the affine invariance condition. Depending on the distribution of data points, the simplicial depth has completely different characteristics. For a *Lebesgue-continuous* distribution [46], the simplicial depth changes continuously (Theorem 2 in [54]),

decreases monotonously on the rays, and has a unique central region [54, 66]. Furthermore, the contours defined by simplicial depth are nested (Theorem 3 in [54]). However, if the distribution is discrete, these characteristics are not necessary applicable [90]. Unlike the contours of halfspace depth which are convex, the contours of the simplicial depth are only starshaped (Section 2.3.3 in [66]). It has been proven that the breakdown point of the simplicial median is always worse than the breakdown point of halfspace median [28]. The behaviour of simplicial depth contours is not as nice as the behaviour of half-space depth contours [47, 63]. As an example, Figure 3.8 illustrates that the simplicial depth contours may not be nested. It can be seen that the contour enclosing all points of depth 10/20 and up is not surrounded by the contour enclosing depth of 8/20 and up.

Simplicial depth is widely studied in the literature. Some of the results regarding simplicial depth can be listed as follows.

- The simplicial depth of a query point in $\mathbb{R}^2$ can be computed using an optimal algorithm which takes $\Theta(n \log n)$ time [9, 12, 51].
- The simplicial depth of a point in $\mathbb{R}^3$ can be computed in $O(n^2)$, and in $\mathbb{R}^4$ the fastest known algorithm needs $O(n^4)$ time [29]. In the higher dimension d, no better algorithm is known than the brute force method with $O(n^{d+1})$ time.
- The maximum simplicial depth in $\mathbb{R}^2$ can be computed in $O(n^4 \log(n))$ [9].

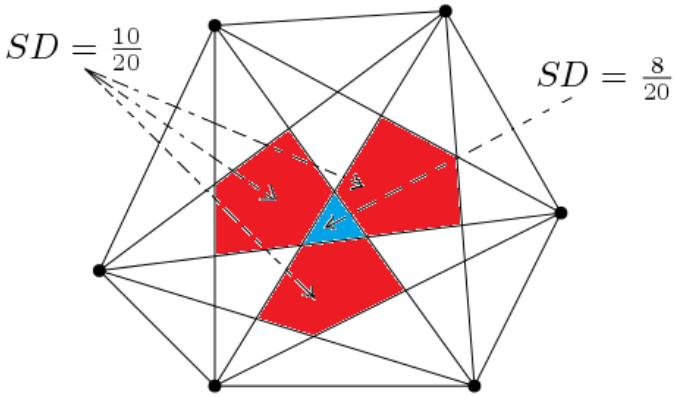

Figure 3.8: Simplicial depth contours

3.4 β-skeleton Depth

To introduce the β-skeleton depth, first, we need to define the β-influence region.

Definition 3.4.1. *For $1 \leq \beta \leq \infty$, the β-influence region of vectors x_i and x_j ($S_\beta(x_i, x_j)$) is defined as follows:*

$$S_\beta(x_i, x_j) = B(c_i, r) \cap B(c_j, r), \tag{3.3}$$

where $r = \frac{\beta}{2}\|x_i - x_j\|$, $c_i = \frac{\beta}{2}x_i + (1 - \frac{\beta}{2})x_j$, and $c_j = (1 - \frac{\beta}{2})x_i + \frac{\beta}{2}x_j$. In the case of $\beta = \infty$, the β-skeleton influence region is defined as a slab determined by two halfspaces perpendicular to the line segment $\overline{x_i x_t}$ at the end points. Figure 3.9 shows the β-skeleton influence regions for different values of β.

Note 3.4.1. *For $1 \leq \beta \leq \infty$ in literature, the ball based version of $S_\beta(x_i, x_j)$*

is also defined. In this case, the $S_\beta(x_i, x_j)$ is given by the union of the balls, instead of the intersection of them in equation 3.3. For example, the hatching area in Figure 3.9 denotes the ball based version of the $S_2(x_1, x_2)$. Since the definition of the β-skeleton depth is given based on the lune based $S_\beta(x_i, x_j)$ alone [64], by $S_\beta(x_i, x_j)$ we only mean its lune based version hereafter in this study.

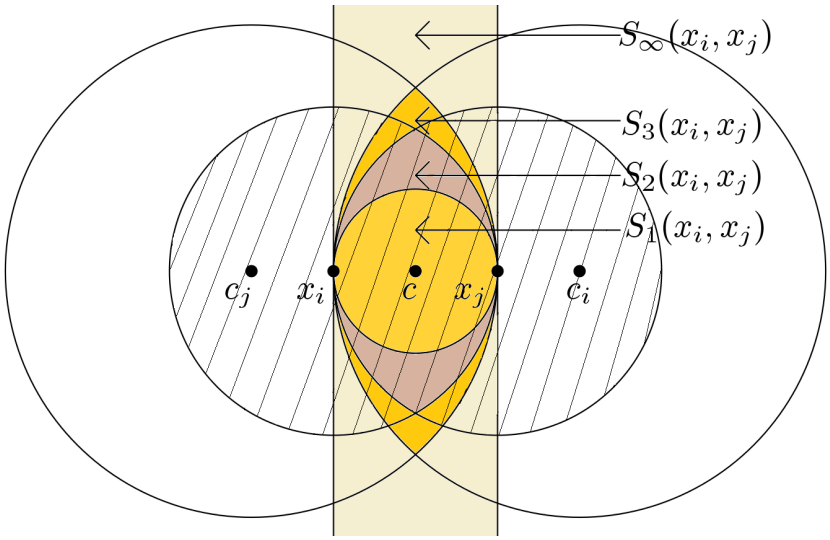

Figure 3.9: The β-influence regions defined by x_i and x_j for β=1, 2, 3, and ∞.

Definition 3.4.2. *For parameter $1 \leq \beta \leq \infty$ and $S = \{x_1, ..., x_n\} \subset \mathbb{R}^d$, the β-skeleton depth of a query point $q \in \mathbb{R}^d$ with respect to S, is defined as the proportion of the β-influence regions that contain q. Using notation SkD_β*

for beta-skeleton depth, this definition can be presented by Equation (3.4).

$$SkD_{\beta}(q;S) = \frac{1}{\binom{n}{2}} \sum_{1 \le i < j \le n} I(q \in S_{\beta}(x_i, x_j)), \tag{3.4}$$

where $1/\binom{n}{2}$ *is the normalization factor.*

It can be verified that $q \in S_{\beta}(x_i, x_j)$ is equivalent to the inequality of $\frac{\beta}{2}\|x_i - x_j\| \geq \max\{\|q - c_i\|, \|q - c_j\|\}$. The straightforward algorithm for computing the β-skeleton depth of $q \in \mathbb{R}^d$ takes $O(dn^2)$ time because the inequality should be checked for all $1 \leq i, j \leq n$ [88, 58]. The β-skeleton depth is a family of statistical depth functions including the spherical depth when $\beta = 1$ [38], and the lens depth when $\beta = 2$ [58].

It is proved that the β-skeleton depth functions satisfy the data depth framework provided by Zuo and Serfling [90] because these depth functions are monotonic (Theorem 3, [88]), maximized at the center, and vanishing at infinity (Theorem 5, [88]). The β-skeleton depth functions are also orthogonally (affinely) invariant if the Euclidean (Mahalanobis) distance is used to construct the influence regions of β-skeleton depth influence regions (Theorem 1, [88]). The breakdown point of β-skeleton median is at least $1-(1/\sqrt{2})$ (Theorem 12, [88]). Regarding the geometric properties of β-skeleton depth, some of the results are as follows. The depth regions with the same depth value are not necessarily connected (see the regions with the depth of 5/6 in Figures 4.4 and 4.5). However, the depth regions are nested (Lemma 3, [88])

which means that the contour with depth of α is geometrically surrounded by the contour with depth of γ, where $\alpha > \gamma$. For example, in Figure 3.10, the contour with the depth of 2/3 is surrounded by the contour with the depth of 1/3. Another property is that the only depth regions which are convex are the central regions (see Theorem 4.2.1). We explore some other geometric properties related to the β-skeleton depth in Chapter 4.

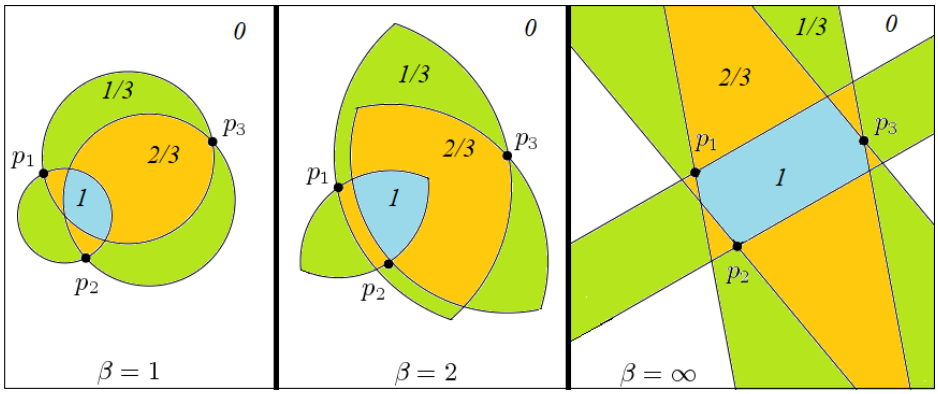

Figure 3.10: Partitions of the plane by β-skeleton depth with respect to $\{p_1, p_2, p_3\}$.

3.4.1 Spherical Depth and Lens Depth

As discussed above, the β-skeleton depth includes the spherical depth when $\beta = 1$, and the lens depth when $\beta = 2$. From Equations (3.3) and (3.4), the definitions of spherical depth ($SphD$) and lens depth (LD) of a query point q with respect to a given data set S in $\mathbb{R}^d$ are as follows:

$$SphD(q; S) = SkD_1(q; S) = \frac{1}{\binom{n}{2}} \sum_{1 \le i < j \le n} I(q \in Sph(x_i, x_j)) \qquad (3.5)$$

$$LD(q;S) = SkD_2(q;S) = \frac{1}{\binom{n}{2}} \sum_{1 \leq i < j \leq n} I(q \in L(x_i, x_j)), \tag{3.6}$$

where the influence regions $Sph(x_i, x_j)$ and $L(x_i, x_j)$ are equivalent to $S_1(x_i, x_j)$ and $S_2(x_i, x_j)$, respectively. The influence region of the spherical depth is also know as the *Gabriel sphere*.

Chapter 4

Geometric Results in $\mathbb{R}^2$

In this chapter we discuss the β-skeleton depth in $\mathbb{R}^2$ from a geometric point of view. The geometric results provide some guidance to the algorithms for computing β-skeleton depth. As an example, in Section 4.1, we explore that computing the entire arrangement of β-influence regions in $\mathbb{R}^2$ is not an efficient approach to compute the planar β-skeleton depth. Given a set of collinear points in $\mathbb{R}^2$, we compute the combinatorial complexity (CC) of the arrangement of β-influence regions. Some geometric properties of the β-skeleton depth are also explored in this chapter.

4.1 Combinatorial Complexity

For the β-influence regions obtained from a set of collinear points in $\mathbb{R}^2$, we present exact bounds for the number of edges, faces, and vertices in the

corresponding arrangement.

Definition 4.1.1. *The CC of an arrangement in* $\mathbb{R}^2$ *is equal to the total number of faces, edges, and vertices (intersection points and data points) in the arrangement.*

Example 4.1.1. *For a set of four collinear points* $p_1, p_2, p_3,$ *and* p_4 *in* $\mathbb{R}^2$*, consider the arrangement of corresponding* β*-influence regions (*$1 \leq \beta < \infty$*). The CC of this arrangement is equal to* 34 *because the arrangement includes the total number of* 12 *faces,* 16 *edges, and* 6 *vertices. For the case of* $\beta = 1$*, see Figure 4.1.*

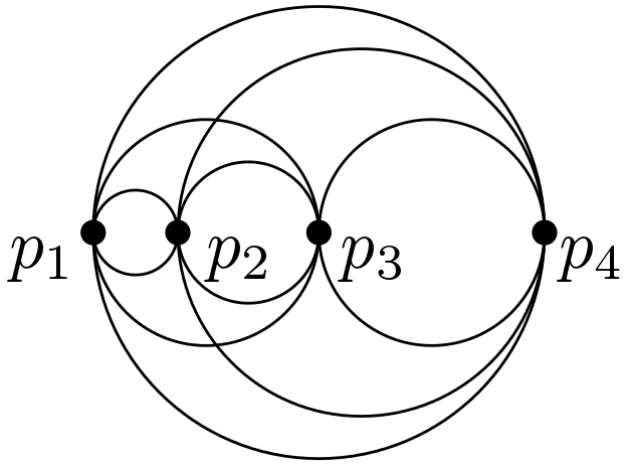

Figure 4.1: Arrangement of β-influence regions ($\beta = 1$) for collinear points

Theorem 4.1.1. *For the arrangement of all Gabriel circles obtained from* n *distinct collinear points in* $\mathbb{R}^2$*,*

$$CC = \frac{1}{4}n^4 - \Theta(n^3).$$

Proof. We construct the arrangement of Gabriel circles incrementally, and define some strategies to count the number of faces, edges, and vertices in the arrangement. Starting with the two leftmost points in the data set, we have one Gabriel circle to consider. In this step, there are only two faces (inside the circle and outside the circle), two edges, and two vertices. Henceforward we add the data points one by one from left to right. We count the number of created faces, edges, and vertices after adding each Gabriel circle obtained from the new point and any previously added data points. We write the numbers of new faces, edges, and vertices in rows corresponding to faces, edges, and vertices, respectively. The new cells are obtained from the intersection of recently added circle and the cells in the previous arrangement. Finally, it is enough to sum up all obtained corresponding numbers in order to get the total number of all faces, edges, and vertices. These strategies are represented in the triangular forms in Tables 4.1, 4.2, and 4.3. These representations help to obtain a general formula for every element of the tables. Figure 4.2 illustrates how to obtain the numbers for $n = 3$ from the previous step (see the rows $n = 2$ and $n = 3$ in Tables 4.1, 4.2, and 4.3). Note that we respectively define $f_k(n)$, $e_k(n)$, and $v_k(n)$ to be the number of recently created faces, edges, and vertices after including the new Gabriel circle $S_1(n^{th}point,\ (n-k)^{th}point)$ in the previously updated arrangement. We define a function δ_{ij}^r which helps us to formulate $f_k(n)$, $e_k(n)$, and $v_k(n)$

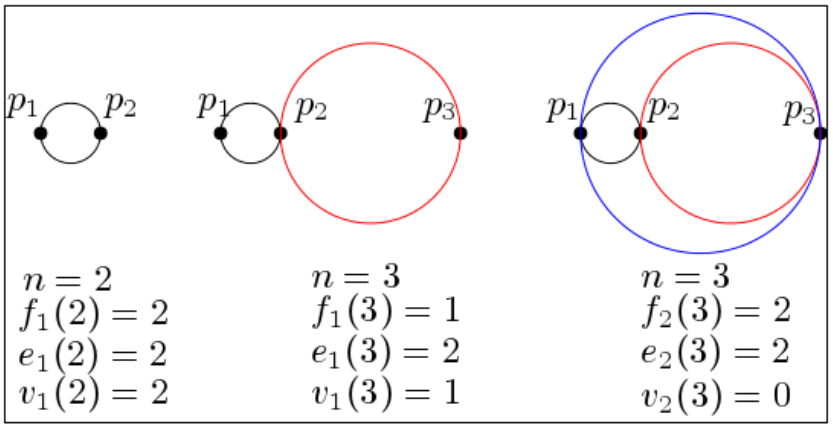

Figure 4.2: Visualization of rows $n = 2$ and $n = 3$ of Table 4.1, Table 4.2, and Table 4.3

$\|S\|$	Faces added per data point in previous arrangement														
2							2								
3						1			2						
4					1			4			2				
5				1			6			6			2		
6			1			8			10			8			2
7		1			10			14			14			10			2
8	1			12			18			20			18			12			2
n	$f_1(n)$			...					$f_k(n)$					...					$f_{n-1}(n)$

Table 4.1: The number of faces in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points.

in Tables 4.1, 4.2, and 4.3, respectively.

$$\delta_{ij}^{r} = \begin{cases} r & \text{if } i = j \\ i & \text{if } i < j \\ 0 & \text{otherwise} \end{cases}$$

$\|S\|$	Vertices added per data point in previous arrangement														
2							2								
3						2			2						
4					2		4		2						
5				2		6			6		2				
6			2		8		10		8		2				
7		2		10		14			14		10		2		
8	2		12		18		20		18		12		2		
n	$e_1(n)$		...				$e_k(n)$				...				$e_{n-1}(n)$

Table 4.2: The number of edges in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points.

$\|S\|$	Edges added per data point in previous arrangement														
2							2								
3						1			0						
4					1		2		0						
5				1		4			4		0				
6			1		6		8		6		0				
7		1		8		12			12		8		0		
8	1		10		16		18		16		10		0		
n	$v_1(n)$		...				$v_k(n)$				...				$v_{n-1}(n)$

Table 4.3: The number of vertices in the incremental construction of the arrangement of Gabriel circles obtained from collinear data points.

The elements $f_k(n)$, $e_k(n)$, and $v_k(n)$ can be presented by following equations.

$$f_k(n) = \begin{cases} f_k(n-1) + 2\delta^1_{(k-1)(n-2)} & \text{if } 2 \le k \le n-1 \text{ and } n \ge 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 1 & \text{if } k = 1 \text{ and } n \ge 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.1}$$

$$e_k(n) = \begin{cases} e_k(n-1) + 2\delta^1_{(k-1)(n-2)} & \text{if } 2 \le k \le n-1 \text{ and } n \ge 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 2 & \text{if } k = 1 \text{ and } n \ge 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$$v_k(n) = \begin{cases} v_k(n-1) + 2\delta^0_{(k-1)(n-2)} & \text{if } 2 \le k \le n-1 \text{ and } n \ge 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 1 & \text{if } k = 1 \text{ and } n \ge 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

We employ the Telescoping Substitution to solve the recurrences in Equations (4.1), (4.2), and (4.3) as follows. For $2 \le k \le n-1$ and $n \ge 3$,

$$\begin{aligned} f_k(n) &= f_k(n-1) + 2\delta^1_{(k-1)(n-2)} \\ &= f_k(n-2) + 2\delta^1_{(k-1)(n-3)} + 2\delta^1_{(k-1)(n-2)} \\ &= f_k(n-3) + 2\delta^1_{(k-1)(n-4)} + 2\delta^1_{(k-1)(n-3)} + 2\delta^1_{(k-1)(n-2)} \\ &\vdots \\ &= f_k(k) + 2\left(\delta^1_{(k-1)(k-1)} + \delta^1_{(k-1)(k)} + \ldots + \delta^1_{(k-1)(n-3)} + \delta^1_{(k-1)(n-2)}\right) \\ &= 0 + 2\left(1 + \sum_{i=1}^{n-k-1} \delta^1_{(k-1)(k+i-1)}\right) = 2\left(1 + (k-1)(n-k-1)\right). \end{aligned}$$

Therefore,

$$f_k(n) = \begin{cases} 2(1+(k-1)(n-k-1)) & \text{if } 2 \leq k \leq n-1 \text{ and } n \geq 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 1 & \text{if } k = 1 \text{ and } n \geq 3 \\ 0 & \text{otherwise.} \end{cases} \tag{4.4}$$

Similarly, we can solve $e_k(n)$ and $v_k(n)$ and obtain the following relations.

$$e_k(n) = \begin{cases} 2(1+(k-1)(n-k-1)) & \text{if } 2 \leq k \leq n-1 \text{ and } n \geq 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 2 & \text{if } k = 1 \text{ and } n \geq 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$$v_k(n) = \begin{cases} 2(k-1)(n-k-1) & \text{if } 2 \leq k \leq n-1 \text{ and } n \geq 3 \\ 2 & \text{if } k = 1 \text{ and } n = 2 \\ 1 & \text{if } k = 1 \text{ and } n \geq 3 \\ 0 & \text{otherwise} \end{cases} \tag{4.6}$$

To compute the CC of the arrangement, we need to calculate C_f, C_e, and C_v representing the total numbers of faces, edges, and vertices, respectively. These values can be similarly computed using Equations (4.4), (4.5), and (4.6), respectively. We provide the computation of C_f in the following. How-

ever, to avoid repetitions, we omit the computations of C_e and C_v, and present only their final values.

$$\begin{aligned}
C_f &= \sum_{k,n} f_k(n) = 2 + \sum_{m=3}^{n}(1 + \sum_{k=2}^{m-1} f_k(m)) \\
&= 2 + \sum_{m=3}^{n}(1 + \sum_{k=2}^{m-1} 2((k-1)(m-k-1)+1))) \\
&= 2 + (n-2) + 2\sum_{m=3}^{n}\sum_{k=2}^{m-1}(k-1)(m-k-1) + 2\sum_{m=3}^{n}\sum_{k=2}^{m-1} 1 \\
&= n + \frac{2}{6}\sum_{m=3}^{n}(m^3 - 6m^2 + 11m - 6) + 2\sum_{m=3}^{n}(m-2) \\
&= n + (\frac{2}{6})(\frac{1}{4})(n^4 - 6n^3 + 11n^2 - 6n) + 2\frac{(n-2)(n-1)}{2} \\
&= \frac{1}{12}\left(n^4 - 6n^3 + 23n^2 - 30n + 24\right).
\end{aligned}$$

$$C_e = \sum_{k,n} e_k(n) = 2 + \sum_{m=3}^{n}(2 + \sum_{k=2}^{m-1} e_k(m)) = \frac{1}{12}(n^4 - 6n^3 + 23n^2 - 18n).$$

$$C_v = \sum_{k,n} v_k(n) = 2 + \sum_{m=3}^{n}(1 + \sum_{k=2}^{m-1} v_k(m)) = \frac{1}{12}(n^4 - 6n^3 + 11n^2 + 6n).$$

The proof is complete because

$$CC = C_f + C_e + C_v = \frac{1}{4}(n^4 - 6n^3 + 19n^2 - 14n + 8) = \frac{1}{4}n^4 - \Theta(n^3).$$

□

Note 4.1.1. *Recalling that the Gabriel circles and β-influence regions are*

equivalent when $\beta = 1$, one can generalize Lemma 4.1.1 by considering β-influence regions ($1 \leq \beta < \infty$) instead of Gabriel circles. This generalization can be made because the corresponding combinatorics does not change for β-influence regions if $1 \leq \beta < \infty$. However, the case of $\beta = \infty$ does not follow this generalization. In this case, $CC = \Theta(n)$ because $C_f = n$ (including one face outside all edges and $n-1$ faces among the edges), $C_e = n$ (including n distinct parallel lines passing through data points), and $C_v = n+2$ (including n data points and 2 intersection points at infinity). We recall that for n distinct collinear points in $\mathbb{R}^2$, the β-influence regions form some parallel slabs if $\beta = \infty$.

Lemma 4.1.1. *Every distinct pair of β-influence regions cut each other in at most 4 points.*

Proof. Every β-influence region can be represented by a pair of circular arcs. For $\beta = 1$, these circular arcs are some half circles, whereas for $\beta > 1$, they are smaller than half circles. We prove the lemma by considering two cases: $\beta = 1$ and $\beta > 1$. For the case $\beta = 1$, the proof is trivial because two distinct 1-influence regions (i.e. circles) cut each other in at most two points. For the case $\beta > 1$, suppose that $S_\beta(a, b)$ and $S_\beta(c, d)$ are two arbitrary and distinct β-influence regions. Figure 4.3 is an illustration of the β-influence regions ($\beta = 3/2$) and their corresponding circular arcs. Suppose that $S_\beta(a, b)$, as a pair of circular arcs EaF and FbE, arbitrarily cuts $S_\beta(c, d)$. Each one of EaF and FbE is smaller than its corresponding half circle. This implies that, after cutting the boundary of $S_\beta(c, d)$ in at most two points, none of EaF

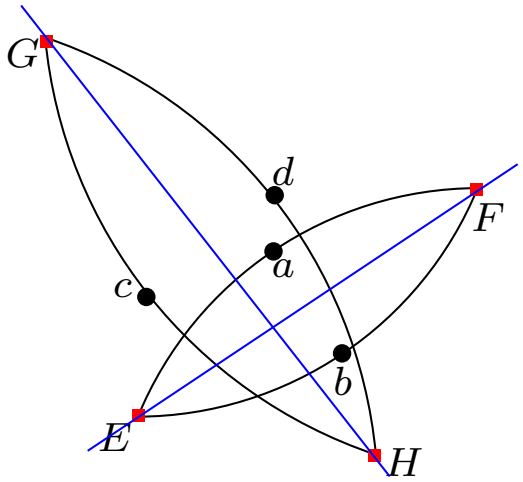

Figure 4.3: Two arbitrarily crossed β-influence regions, $\beta = 3/2$.

and FbE would turn back towards the boundary of $S_\beta(c, d)$. As such, two β-influence regions $S_\beta(a, b)$ and $S_\beta(c, d)$ cut each other in at most 4 points.

□

Lemma 4.1.2. *The trivial upper bound for the CC of the arrangement of all β-influence regions obtained from n arbitrarily distributed data points in $\mathbb{R}^2$ is $O(n^4)$.*

Proof. From Lemma 4.1.1, the boundaries of every two β-influence regions intersect each other in at most 4 points. It means that we have at most $4\binom{\binom{n}{2}}{2}$ intersection points in the arrangement of the $\binom{n}{2}$ β-influence regions. Since every arrangement is a representation of a planar graph, we can use Euler's Formula $C_v - C_e + C_f = 2$ (Theorem 2.1.1) in planar graphs to compute the number of faces, edges, and vertices in the arrangement. The number of vertices and the number of edges in the planar graphs are related to each

other by the inequality $C_e \leq 3C_v - 6$ (see Theorem 9.7 in [8]). Considering the intersection points from the above discussion and the number of data points, C_v and consequently C_e, C_f, and CC can be computed as follows:

$$C_v \leq 4\binom{\binom{n}{2}}{2} + n = \frac{1}{2}(n^4 - 2n^3 - n^2 + 4n)$$
$$C_e \leq 3C_v - 6 \leq \frac{3}{2}(n^4 - 2n^3 - n^2 + 4n) - 6 = \frac{3}{2}(n^4 - 2n^3 - n^2 + 4n - 4)$$
$$C_f = 2 + C_e - C_v \leq 2 + (2C_v - 6) = 2C_v - 4 \leq n^4 - 2n^3 - n^2 + 4n - 4$$
$$CC = C_f + C_e + C_v \leq (3n^4 - 6n^3 - 3n^2 - 12n - 10) \in O(n^4).$$

□

Note 4.1.2. *Lemma 4.1.1 indicates that the trivial upper bound $O(n^4)$ for the CC related to the arrangement of β-influence regions is achievable.*

Conjecture 4.1.1. *The collinear configuration of n planar points minimizes the number of intersections (and consequently, edges and faces) among the corresponding β-influence regions.*

Proving Conjecture 4.1.1 would imply a lower bound for the CC of the arrangement of β-influence regions for every arbitrary set of points in $\mathbb{R}^2$. Furthermore, it would also imply that the obtained upper bound in Lemma 4.1.2 is optimal. Without loss of generality, it can be assumed that $\beta = 1$.

4.2 Geometric Properties of β-skeleton Depth

In this section, some geometric properties of the β-skeleton depth are investigated. Among all of the β-influence regions, we only consider the case $\beta = 1$ (i.e. Gabriel circles) related to the spherical depth. One can easily generalize the results to the other value of β.

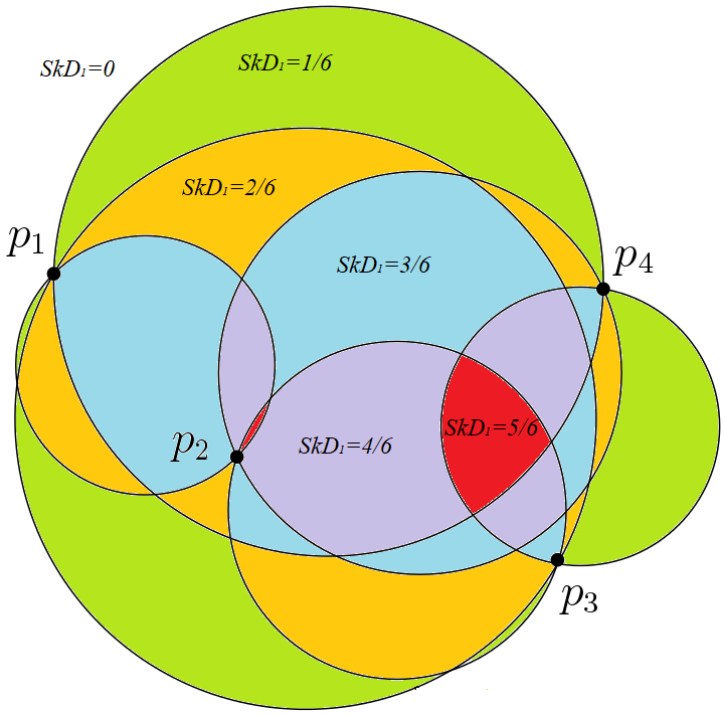

Figure 4.4: Spherical depth of planar points with respect to $\{p_1, p_2, p_3, p_4\}$

As we discussed in Section 3.4, similar to the property of halfspace depth in Section 3.2, the contours of β-skeleton depth are nested. However, we may have more than one contour per depth value. For example, in Figures 4.4 and 4.5, it can be seen that there are two separate contours with the depth of 5/6.

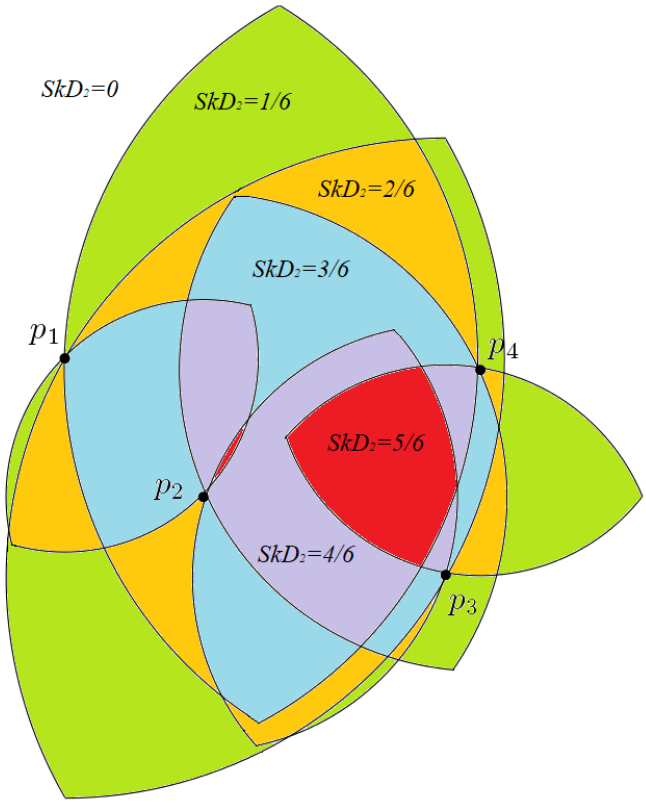

Figure 4.5: Lens depth of planar points with respect to $\{p_1, p_2, p_3, p_4\}$

If there exists more than one contour for a depth value, we have some locally deepest regions in the arrangement. Consequently, the central regions may not be connected (see Figures 4.4 and 4.5). Another geometric property of β-skeleton depth is that the only convex depth regions are the locally deepest regions. These last two properties can be deduced from Lemma 4.2.1 and Theorem 4.2.1.

Lemma 4.2.1. *For every arbitrary pair of neighboring faces*[1] *A and B in*

[1]Two faces are neighbors if they have a common edge in their boundaries.

the arrangement of Gabriel circles obtained from a data set $S \subset \mathbb{R}^2$,

$$\forall (a \in A, b \in B); |\, SkD_{1S}(a) - SkD_{1S}(b)| \geq 1, \tag{4.7}$$

where $SkD_{1S}(a) = \binom{n}{2} SkD_1(a; S)$.

Proof. The proof is immediate from the fact that every edge of the arrangement corresponds to some Gabriel circle G. One of the two faces bounded by this edge is entirely contained in G, and the other face is entirely outside of G. □

Note 4.2.1. *Lemma 4.2.1 implies that the faces with the same depth can be connected only in discrete points.*

Note 4.2.2. *Considering two neighboring regions*[2] *instead of two neighboring faces, one can generalize Lemma 4.2.1. To obtain such generalization, it is enough to apply Lemma 4.2.1 on any arbitrary pair of neighboring faces in the neighboring regions.*

Theorem 4.2.1. *Consider the arrangement of Gabriel circles obtained from data set* S. *A face (region) in this arrangement is locally deepest if and only if it has no concave edge. In other words, a face is locally deepest if and only if it is a convex face.*

Proof. ⇒) Suppose that A is a locally deepest face in the arrangement of Gabriel circles obtained from S. We prove that A has no concave edge. To

[2]Two regions are neighbors if they have a common border.

obtain a contradiction, assume that e is a concave edge of A. Consequently, there exists a neighboring face B, in the arrangement, whose boundary contains e. Lemma 4.2.1 and the characteristics of concave edge in the arrangement of Gabriel circles imply that $SkD_1(B;S) > SkD_1(A;S)$. This result contradicts the assumption that A is locally deepest.

$\Leftarrow$) Suppose that A is a face that has no concave edge in the arrangement of Gabriel circles obtained from S. We prove that A is locally deepest. Assume that the boundary of A is composed of the convex edges $e_1, ...e_k$, and thus A has neighboring faces $B_1, ..., B_m$ $(m \leq k)$. Hence A and every B_i $(i = 1, ..., m)$ have at least one common edge e_j $(j = 1, ..., k)$. From Lemma 4.2.1 and the characteristics of concave edge, it can be seen that

$$\forall B_i;\ SkD_1(A;S) > SkD_1(B_i;S).$$

This means that A is a locally deepest because it is the deepest face among all of its neighboring faces B_i. $\square$

Chapter 5

Algorithmic Results in $\mathbb{R}^2$

The previous best algorithm for computing the β-skeleton depth of a point $q \in \mathbb{R}^d$ with respect to a data set $S = \{x_1..x_n\} \subseteq \mathbb{R}^d$ is the brute force algorithm [89]. This naive algorithm needs to check all of the $\binom{n}{2}$ β-skeleton influence regions obtained from the data points to figure out how many of them contain q. Checking all of such influence regions causes the naive algorithm to take $\Theta(dn^2)$ time. In this chapter, we present an optimal algorithm for computing the planar spherical depth ($\beta = 1$) and a subquadratic algorithm to compute the planar β-skeleton depth when $\beta > 1$. In these algorithms, we need to solve some halfspace and some circular range counting problems, where all of the halfspaces have one common point. The circles also have the same characteristic. Furthermore, computing the planar β-skeleton depth is reduced to a combination of some range counting problems. In the special case $\beta = 1$, we investigate a specialized halfspace range query method that

leads to a $\Theta(n \log n)$ algorithm (Algorithm 5) for β-skeleton depth. Finally, we present a simple and optimal algorithm (Algorithm 7) that computes the planar halfspace depth in $\Theta(n \log n)$ time. This algorithm is similar to the Aloupis's algorithm in [9]. After sorting the points by angle, Aloupis employed the counterclockwise sweeping of a specific halfline. However, in our algorithm, we use the specialized halfspace range query that is explored in Algorithm 5.

5.1 Optimal Algorithm to Compute Planar Spherical Depth

Instead of checking all of the spherical influence regions, we focus on the geometric aspects of such regions in $\mathbb{R}^2$. The geometric properties of these regions lead us to develop an $\Theta(n \log n)$ algorithm for the computation of planar spherical depth of $q \in \mathbb{R}^2$.

Theorem 5.1.1. *For arbitrary points a, b, and t in $\mathbb{R}^2$, $t \in Sph(a, b)$ if and only if $\angle atb \geq \frac{\pi}{2}$.*

Proof. If t is on the boundary of $Sph(a, b)$, *Thales' Theorem*[1] suffices as the proof in both directions. For the rest of the proof, by $t \in Sph(a, b)$ we mean $t \in int\ Sph(a, b)$.

$\Rightarrow$) For $t \in Sph(a, b)$, suppose that $\angle atb < \frac{\pi}{2}$ (proof by contradiction). We

[1]Thales' Theorem also known as the *Inscribed Angle Theorem*: If a, b, and c are points on a circle where $\overline{ac}$ is a diameter of the circle, then $\angle abc$ is a right angle.

continue the line segment $\overline{at}$ to cross the boundary of the $Sph(a,b)$. Let t' be the crossing point (see the left figure in Figure 5.1). Since $\angle atb < \frac{\pi}{2}$, then, $\angle btt'$ is greater than $\frac{\pi}{2}$. Let $\angle btt' = \frac{\pi}{2} + \epsilon_1; \epsilon_1 > 0$. From Thales' Theorem, we know that $\angle at'b$ is a right angle. The angle $tbt' = \epsilon_2 > 0$ because $t \in Sph(a,b)$. Summing up the angles in $\triangle tt'b$, as computed in (5.1), leads to a contradiction. So, this direction of proof is complete.

$$\angle tt'b + \angle t'bt + \angle btt' \geq \frac{\pi}{2} + \epsilon_2 + \frac{\pi}{2} + \epsilon_1 > \pi \tag{5.1}$$

$\Leftarrow$) If $\angle atb = \frac{\pi}{2} + \epsilon_1; \epsilon_1 > 0$, we prove that $t \in Sph(a,b)$. Suppose that $t \notin Sph(a,b)$ (proof by contradiction). Since $t \notin Sph(a,b)$, at least one of the line segments $\overline{at}$ and $\overline{bt}$ crosses the boundary of $Sph(a,b)$. Without loss of generality, assume that $\overline{at}$ is the one that crosses the boundary of $Sph(a,b)$ at the point t' (see the right figure in Figure 5.1). Considering Thales' Theorem, we know that $\angle at'b = \frac{\pi}{2}$ and consequently, $\angle bt't = \frac{\pi}{2}$. The angle $\angle t'bt = \epsilon_2 > 0$ because $t \notin Sph(a,b)$. If we sum up the angles in the triangle $\triangle tt'b$, the same contradiction as in (5.1) will be implied. $\square$

Algorithm 5: Using Theorem 5.1.1, we present an algorithm to compute the spherical depth of a query point $q \in \mathbb{R}^2$ with respect to $S = \{x_1, ..., x_n\} \subseteq \mathbb{R}^2$. This algorithm is summarized in the following steps. The pseudocode of this algorithm is provided at the end of this chapter.

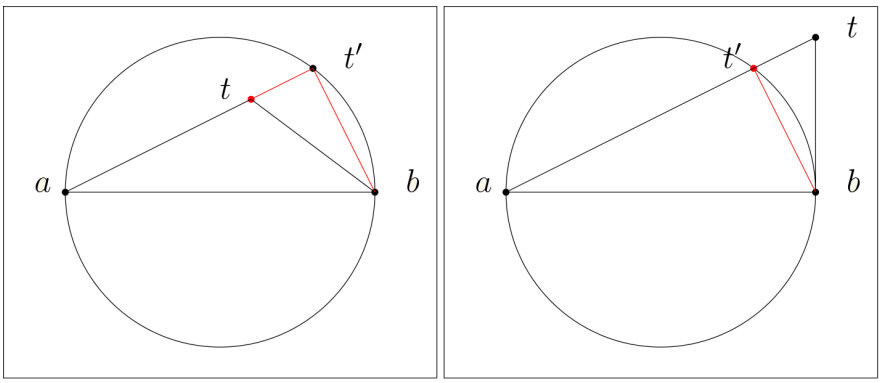

Figure 5.1: Point t and spherical influence region $Sph(a, b)$

- **Translating the points:** Suppose that T is a translation by $(-q)$. We apply T to translate q and all data points into their new coordinates. Obviously, $T(q) = O$.

- **Sorting the translated data points:** In this step we sort the translated data points based on their angles in their polar coordinates. After doing this step, we have S_T which is a sorted array of the translated data points.

- **Calculating the spherical depth:** For the i^{th} element in S_T, we define O_i and N_i as follows:

$$\begin{aligned} O_i &= \left\{ j : x_j \in S_T \, , \frac{\pi}{2} \leq |\theta(x_i) - \theta(x_j)| \leq \frac{3\pi}{2} \right\} \\ N_i &= \{1, 2, ..., n\} \setminus O_i. \end{aligned} \tag{5.2}$$

Thus the spherical depth of q with respect to S, can be computed by:

$$SphD(q;S) = SphD(O;S_T) = \frac{1}{2\binom{n}{2}} \sum_{1 \le i \le n} |O_i|. \tag{5.3}$$

To present a formula for computing $|O_i|$, we define $f(i)$ and $l(i)$ as follows:

$$f(i) = \begin{cases} \min N_i - 1 & \text{if } \frac{\pi}{2} < \theta(x_i) \le \frac{3\pi}{2} \\ \min O_i & \text{otherwise} \end{cases}$$

$$l(i) = \begin{cases} \max N_i + 1 & \text{if } \frac{\pi}{2} < \theta(x_i) \le \frac{3\pi}{2} \\ \max O_i & \text{otherwise.} \end{cases}$$

Figure 5.2 illustrates O_i, N_i, $f(i)$, and $l(i)$ in two different cases. Considering the definitions of $f(i)$ and $l(i)$,

$$|O_i| = \begin{cases} f(i) + (n - l(i) + 1) & \text{if } \frac{\pi}{2} < \theta(x_i) \le \frac{3\pi}{2} \\ l(i) - f(i) + 1 & \text{otherwise.} \end{cases} \tag{5.4}$$

This allows us to compute $|O_i|$ using a pair of binary searches.

Time complexity of Algorithm 5: The first procedure in the algorithm takes $\Theta(n)$ time to translate q and all data points into the new coordinate system. The second procedure takes $\Theta(n \log n)$ time. Due to using binary search for every O_i, the running time of the last procedure is also $\Theta(n \log n)$.

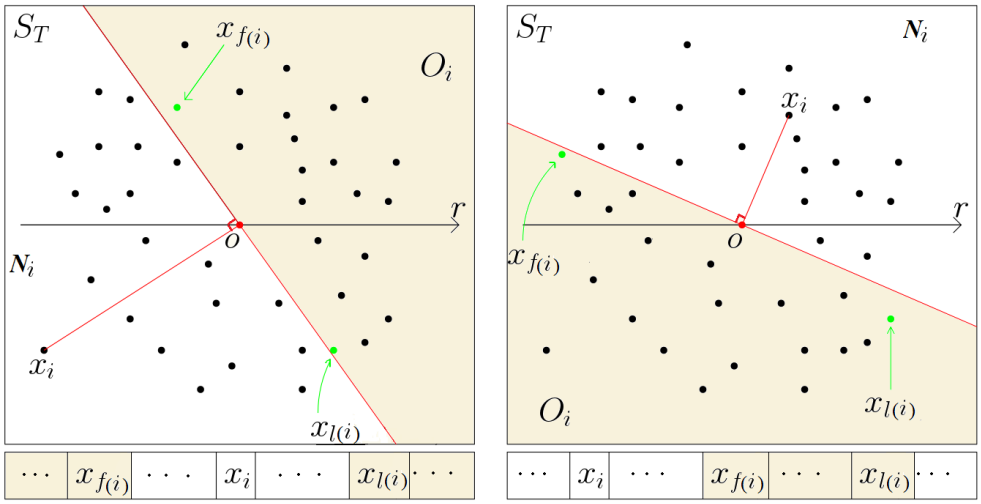

Figure 5.2: $\theta(x_i) \in (\frac{\pi}{2}, \frac{3\pi}{2}]$ (left figure), and $\theta(x_i) \notin (\frac{\pi}{2}, \frac{3\pi}{2}]$ (right figure)

The rest of the algorithm contributes some constant time. In total, the running time of the algorithm is $\Theta(n \log n)$.

Note 5.1.1. ***Coordinate system:*** *In practice it may be preferable to work in the Cartesian coordinate system. Sorting by angle can be done using some appropriate right-angle tests (determinants). Regarding the other angle comparisons, they can be done by checking the sign of dot products.*

5.2 Algorithm to Compute Planar β-skeleton Depth when $\beta > 1$

We recall from the definition of β-influence region that $S_\beta(x_i, x_j); \beta > 1$ forms some lenses, and $S_\infty(x_i, x_j)$ forms some slabs for each pair x_i and x_j in $S \subseteq \mathbb{R}^2$. Using some geometric properties of such lenses and slabs, we prove

Theorem 5.2.1. This theorem along with some results regarding the range counting problems in [4] help us to compute $SkD_\beta(q;S); \beta > 1$ in $O(n^{\frac{3}{2}+\epsilon})$ time, where q and S are in $\mathbb{R}^2$.

Definition 5.2.1. *For an arbitrary non-zero point $a \in \mathbb{R}^2$ and parameter $\beta \geq 1$, $\ell(p)$ is a line that is perpendicular to $\overrightarrow{a}$ at the point $p = p(a,\beta) = (\beta-1)a/\beta$. This line forms two halfspaces $H_o(p)$ and $H_a(p)$. The closed halfspace that includes the origin is $H_o(p)$ and the other halfspace which is open is $H_a(p)$.*

Definition 5.2.2. *For a disk $B(c,r)$ with the center $c = c(a,\beta) = \beta a/2(\beta-1)$ and radius $r = \|c\|$, $B_o(c,r)$ is the intersection of $H_o(p)$ and $B(c,r)$, and $B_a(c,r)$ is the intersection of $H_a(p)$ and $B(c,r)$, where $\beta > 1$ and a is an arbitrary non-zero point in $\mathbb{R}^2$. For the case of $\beta = 1$, $B(c,r) = \emptyset$.*

Figure 5.3 is an illustration of these definitions for different values of parameter β.

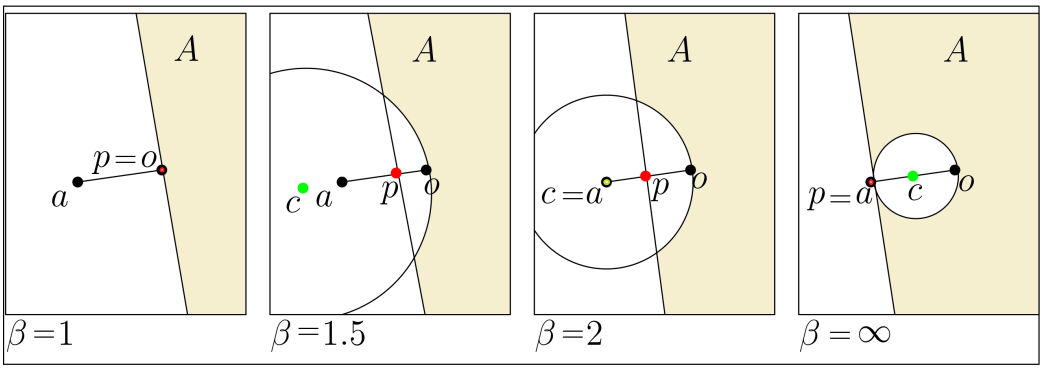

Figure 5.3: $H_o(p)$ and $B(c,r)$ defined by $a \in \mathbb{R}^2$ for $\beta = 1,\ 1.5,\ 2,$ and $\beta = \infty$, where $A = H_o(p) \setminus \{intB_o(c,r)\}$

Theorem 5.2.1. *For non-zero points $a, b \in \mathbb{R}^2$, and parameter $\beta > 1$, $b \in H_o(p) \setminus \{intB_o(c,r)\}$ if and only if the origin $O = (0,0)$ is contained in $S_\beta(a,b)$, where $c = \beta a/(2(\beta-1))$, $r = \|c\|$, and $p = (\beta-1)a/\beta$.*

Proof. First, we show that $B_o(c,r)$ is a well-defined set meaning that $\ell(p)$ intersects $B(c,r)$. We compute $d(c,\ell(p))$, the distance of c from $\ell(p)$, and prove that this value is not greater than r. It can be verified that $d(c,\ell(p)) = d(c,p)$. Let $k = \beta/(2(\beta-1))$; the following calculations complete this part of the proof.

$$\begin{aligned} d(c,p) &= d(\frac{\beta a}{2(\beta-1)}, \frac{(\beta-1)a}{\beta}) = d(ka, \frac{1}{2k}a) \\ &= (k - \frac{1}{2k})\sqrt{({a_x}^2 + {a_y}^2)} = (\frac{2k^2-1}{2k})\|a\| \\ &< \frac{2k^2}{2k}\|a\| = k\|a\| = r. \end{aligned}$$

We recall the definition of β-influence region given by $S_\beta(a,b) = B(c_a, \frac{\beta}{2}\|a-b\|) \cap B(c_b, \frac{\beta}{2}\|a-b\|)$, where $c_a = \frac{\beta}{2}a + (1-\frac{\beta}{2})b$ and $c_b = \frac{\beta}{2}b + (1-\frac{\beta}{2})a$. Using this definition, following equivalencies can be derived from $O \in S_\beta(a,b)$.

$$\begin{aligned} O \in S_\beta(a,b) &\Leftrightarrow \frac{\beta\|a-b\|}{2} \geq max\{\|c_a\|, \|c_b\|\} \\ &\Leftrightarrow \beta\|a-b\| \geq max\{\|\beta(a-b)+2b\|, \|\beta(b-a)+2a\|\} \\ &\Leftrightarrow \beta^2\|a-b\|^2 \geq max\{\|\beta(a-b)+2b\|^2, \|\beta(b-a)+2a\|^2\} \\ &\Leftrightarrow 0 \geq max\{b^2(1-\beta) + \beta\langle a,b\rangle, a^2(1-\beta) + \beta\langle a,b\rangle\}. \end{aligned}$$

By solving these inequalities for $(\beta-1)/\beta$ which is equal to $1/2k$, we obtain Equation (5.5).

$$\frac{1}{2k} \geq max\left\{\frac{\langle a,b\rangle}{\|a\|^2}, \frac{\langle a,b\rangle}{\|b\|^2}\right\} \tag{5.5}$$

For a fixed point a, the inequalities in Equation (5.5) determine one halfspace and one disk given by (5.6) and (5.7), respectively.

$$\frac{1}{2k} \geq \frac{\langle a,b\rangle}{\|a\|^2} \Leftrightarrow \langle a,b\rangle \leq \frac{1}{2k}\|a\|^2. \tag{5.6}$$

$$\begin{aligned} \frac{1}{2k} \geq \frac{\langle a,b\rangle}{\|b\|^2} &\Leftrightarrow b^2 - 2k\langle a,b\rangle \geq 0 \\ &\Leftrightarrow b^2 - 2k\langle a,b\rangle + k^2a^2 \geq k^2a^2 \\ &\Leftrightarrow (b-ka)^2 \geq (k\|a\|)^2. \end{aligned} \tag{5.7}$$

The proof is complete because for a point a, the set of all points b containing in the feasible region defined by Equations (5.6) and (5.7) is equal to $H_o(p) \setminus \{intB_o(c,r)\}$. □

Proposition 5.2.1. *For $\beta = 2+\sqrt{2}$, the boundary of the given halfspace in* (5.6) *passes through the center of the given disk in* (5.7).

Proof. It is enough to substitute b in the given halfspace with ka which is the center of the given disk.

$$\begin{aligned} \langle a,(ka)\rangle \leq \frac{1}{2k}\|a\|^2 \Rightarrow k\|a\|^2 \leq \frac{1}{2k}\|a\|^2 \Rightarrow 2k^2 \leq 1 \\ \Rightarrow 2(\frac{\beta}{2(\beta-1)})^2 \leq 1 \Rightarrow \beta = 2+\sqrt{2} \end{aligned}$$

□

Algorithm 6: Using Theorem 5.2.1, we present an algorithm to compute the β-skeleton depth of $q \in \mathbb{R}^2$ with respect to $S = \{x_1, ..., x_n\} \subseteq \mathbb{R}^2$. This algorithm is summarized in two steps. The pseudocode of this algorithm can be found at the end of this chapter.

- **Translating the points:** This step is exactly the same step as in Algorithm 5.

- **Calculating the β-skeleton depth:** Suppose that $a = (a_x, a_y)$ is an element in S' (translated S). We consider a disk and a line as follows:

$$B(c, r) : (x - ka_x)^2 + (y - ka_y)^2 = (k\|a\|)^2$$

$$\ell(p) : a_x x + a_y y = \frac{1}{2k}\|a\|,$$

  where k, c, r, and p are defined in Theorem 5.2.1. From Theorem 1.2 in [4], we can compute $|H_o(p)|$ with $O(n)$ storage, $O(n \log n)$ expected preprocessing time, and $O(n^{\frac{1}{2}+\epsilon})$ query time, where $|H_o(p)|$ is the number of all elements of S' that are contained in $H_o(p)$. For the elements of S', $|intB_o(c, r)|$ which is defined as the number of elements containing in the interior of $B_o(c, r)$ can also be computed with the same storage, expected preprocessing time, and query time. We recall that $B_o(c, r)$ is the intersection of halfspace $H_o(p)$ and disk $B(c, r)$, where p, c, and r are some functions of a. Finally, $SkD_\beta(q, S)$ which is equal

to $SkD_\beta(O; S')$ can be computed by Equation (5.8).

$$SkD_\beta(q, S) = \frac{1}{2\binom{n}{2}} \left(\sum_p |H_o(p)| - \sum_c |intB_o(c, r)| \right) \tag{5.8}$$

Referring to Definitions 5.2.1 and 5.2.2, $H_o(p)$ and $B_o(c, r)$ can be computed in constant time.

Theorem 5.2.1 and Algorithm 6 are valid for $\beta > 1$. However, the case $\beta = 1$ (Algorithm 5 for spherical depth) can also be included in this result. In this case, $\forall c; B(c, r) = \emptyset$, and consequently, $|intB_o(c, r)| = 0$. Therefore, $SkD_1(q; S)$ which is equal to $SkD_1(O; S')$ can be computed by:

$$SkD_1(q; S) = \frac{1}{2\binom{n}{2}} \sum_p |H_o(p)|. \tag{5.9}$$

Time complexity of Algorithm 6: The translating procedure as is discussed in Algorithm 5, takes $O(n)$ time. With the $O(n \log n)$ expected preprocessing time, the second procedure takes $O(n^{\frac{3}{2}+\epsilon})$ time. In this procedure, the loop iterates n times, and the range counting algorithms take $O(n^{\frac{1}{2}+\epsilon})$ time. The expected preprocessing time $O(n \log n)$ is required to obtain a data structure for the aforementioned range counting algorithms. The rest of the algorithm takes some constant time per loop iteration, and therefore the total expected running time of the algorithm is $O(n^{\frac{3}{2}+\epsilon})$.

5.3 Algorithm to Compute Planar Halfspace Depth

To compute the halfspace depth of query point q with respect to data set S, we need to find the minimum portion of data points separated by a halfspace through q. In this section, we develop an optimal algorithm to compute the planar halfspace depth of a query point. Identical to Aloupis' [9] algorithms, our algorithm takes $\Theta(n \log n)$ time. After sorting data points by angle, Aloupis employed the counterclockwise sweeping of a specific halfline. However, to obtain our algorithm, we reuse most of Algorithm 5, and employ the specialized halfspace range counting that is explored in Section 5.1.

Algorithm 7: Suppose that $S = \{x_1, ..., x_n\} \subset \mathbb{R}^2$ and $q \in \mathbb{R}^2$ are given. We summarize the algorithm in the following steps.

- **Translating the points:** This step is the same as the first step in Algorithm 5.
- **Computing the halfspaces** In Equation (3.1), it is not practical to compute $|S \cap H|$ for all halfspaces H that pass through the query point. Instead of considering all of the halfspaces, we define $\mathbb{H}$ to be a finite set of the desired halfspaces such that we can obtain all possible values of $|S \cap H|$ if $H \in \mathbb{H}$. Computation of $\mathbb{H}$ can be done as follows:

(*i*) Project all of the nontrivial[2] elements of S' (translated S) on the unit circle $C(O, 1)$.

(*ii*) Construct $P = \{\pm x'_{ip}\}$, where x'_{ip} is generated in (*i*).

(*iii*) Using an $O(n \log n)$ sorting algorithm, sort the elements of P by angle in counterclockwise order.

(*iv*) Remove the duplicates in the sorted P.

(*v*) Let m_i be the middle point of each pair of successive elements in the sorted array in (*iii*). Suppose that ℓ_i is the line that passes through the points $O = (0, 0)$ and m_i. Each line ℓ_i forms two halfspaces h_{i1} and h_{i2}. As such, $\mathbb{H}$ can be defined by:

$$\mathbb{H} = \{h_{ij}; 1 \leq i \leq n, j \in \{1, 2\}\}.$$

- **Calculating the halfspace depth:** Similar to the computation of $|O_i|$ in Equation (5.4), a pair of binary searches can be applied to compute each of $|h_{ij} \cap S'|$, where $1 \leq i \leq n$ and $j \in \{1, 2\}$. Therefore, $HD(q; S)$ which is equal to $HD(O; S')$ can be computed by:

$$HD(q; S) = \frac{2}{n}\left(\min_{ij}\{|h_{ij} \cap S'|; h_{ij} \in \mathbb{H}\} + |trivials|\right).$$

Time complexity of Algorithm 7: Referring to the analysis of Algorithm 5, the Translating procedure takes $\Theta(n)$ time. Sorting the elements

[2] $a \in \mathbb{R}^2$ is nontrivial if $\|a\| > 0$.

of P causes the second procedure to take $\Theta(n \log n)$ time. The rest of work in the second procedure takes some linear time. Finally, the depth calculation procedure takes $O(n \log n)$ because for every element in $\mathbb{H}$, two binary searches are called. Since $\mathbb{H}$ contains at most $2n$ elements, the outer loop takes $O(n)$. The binary searches take $O(\log n)$. The rest of the algorithm run in some constant time per loop iteration. From the above analysis, the overall running time of this algorithm is $O(n \log n)$.

5.4 Pseudocode

In this section we provide the pseudocode of the presented algorithms in sections 5.1, 5.2, and 5.3. First, we define some procedures which are used in the depth calculation algorithms.

Algorithm 1 Translation(S, q)

Input: Data set S and Query point q
Output: Translated data set S'

1: **for** each $x_i \in S$ **do**
2: $\quad x_i \leftarrow (x_i - q)$
3: **return** S'

Note 5.4.1. *Instead of using polar coordinates in Algorithm 2, sorting can also be done by applying the suggested method in Note 5.1.1.*

Note 5.4.2. *To avoid unusual notations in Algorithm 5, we use the variables a and b instead of $f(i)$ and $l(i)$, respectively in the text.*

Algorithm 2 Sorting-by-angle(S)

Input: Data set S
Output: Sorted array S_T

1: **for** each $x_i = (a_i, b_i) \in S$ **do**
2: $\quad \theta(x_i) \leftarrow \arctan(b_i/a_i)$
3: Using an $O(n \log n)$ sorting algorithm, sort x_i based on $\theta(x_i)$
4: **return** S_T

Algorithm 3 Trivial-and-Nontrivial(S)

Input: Data set S
Output: Partition S into two sets $Trival$ and $NonTrivial$

1: Initialize $Trivial = \emptyset$, $NonTrivial = \emptyset$
2: **for** each $x_i \in S$ **do**
3: $\quad$ **if** $(\|x_i\| = 0)$
4: $\quad\quad Trivial \leftarrow Trivial \cup \{x_i\}$
5: $\quad$ **else**
6: $\quad\quad NonTrivial \leftarrow NonTrivial \cup \{x_i\}$
7: **return** $(Trivial, NonTrivial)$

Algorithm 4 Projection(S)

Input: Data set S
Output: Projection of S on circle $C(O, 1)$

1: Initialize $S_p = \emptyset$
2: **for** each $x_i \in S$ **do**
3: $\quad x_{ip} \leftarrow x_i/\|x_i\|$
4: $\quad S_p \leftarrow S_p \cup \{\pm x_{ip}\}$
5: **return** S_p

Algorithm 5 Computing the planar β-skeleton depth, $\beta = 1$ (spherical depth)

Input: Data set S and Query point q
Output: $SphD(q; S)$

Initialize $SphD(q; S) = 0$
$S' \leftarrow$ Translation(S, q)
$S_T \leftarrow$ Sorting-by-angle(S')
for each $x_i \in S_T$ **do**
    Initialize $a = 0$ and $b = n + 1$
    Using two *binary search calls*, update the values of a and b
    if $(0 < \theta(x_i) \leq \frac{\pi}{2})$
        $a = min\{j : \theta(x_j) - \theta(x_i) \geq \frac{\pi}{2}\}$
        $b = max\{j : \frac{\pi}{2} \leq \theta(x_j) - \theta(x_i) \leq \frac{3\pi}{2}\}$
    else-if $(\frac{\pi}{2} < \theta(x_i) \leq \frac{3\pi}{2})$
        $a = max\{j : \theta(x_i) - \theta(x_j) \geq \frac{\pi}{2}\}$
        $b = min\{j : \theta(x_j) - \theta(x_i) \geq \frac{\pi}{2}\}$
    else
        $a = min\{j : \theta(x_i) - \theta(x_j) \leq \frac{3\pi}{2}\}$
        $b = max\{j : \frac{\pi}{2} \leq \theta(x_i) - \theta(x_j) \leq \frac{3\pi}{2}\}$
    if $(a = 0$ and $b = n + 1)$
        $|O_i| = 0$
    else
        $$|O_i| = \begin{cases} a + (n - b + 1) & ; \frac{\pi}{2} < \theta(x_i) \leq \frac{3\pi}{2} \\ b - a + 1 & ; \text{otherwise.} \end{cases}$$
    $SphD(q; S) \leftarrow SphD(q; S) + \frac{1}{2\binom{n}{2}}|O_i|$
return $SphD(q; S)$

Algorithm 6 Computing the planar β-skeleton depth, $\beta > 1$

Input: Data set S, Query point q, Parameter $\beta > 1$
Output: $SkD_\beta(q; S)$

1: Initialize $SkD_\beta(q; S) = 0$
2: $S' \leftarrow$ Translation(S, q)
3: **for** each $a \in S'$ **do**
4: Using two $O(n^{\frac{1}{2}+\epsilon})$ range counting algorithms, compute $|H_o(p)|$ and $|intB_o(c, r)|$
(The computations of $H_o(p)$ and $intB_o(c, r)$ take constant time)
5: $SkD_\beta(q; S) \leftarrow SkD_\beta(q; S) + \frac{1}{2\binom{n}{2}} (|H_o(p)| - |intB^o(c, r)|)$
6: **return** $SkD_\beta(q; S)$

Algorithm 7 Computing the planar halfspace depth

Input: Data set S, Query point q
Output: $HD(q; S)$

1: Initialize $HD_q = 0$
2: $S' \leftarrow$ Translation(S, q)
3: $(Triv, NTriv) \leftarrow$ Trivial-and-Nontrivial(S')
4: $P \leftarrow$ Projection$(NTriv)$
5: $P_T \leftarrow$ Sorting(P)
6: $A_P \leftarrow$ Removing-Duplicates(P_T)
7: **for** each a_i in A_P **do**
8: $\ell_i \leftarrow$ line passing through $(0, 0)$ and $m_i = (a_i + a_{i+1})/2$
9: $\mathbb{H} \leftarrow$ the set of all halfspaces h_{ij} defined by ℓ_i, where $1 \leq i \leq n, j \in \{1, 2\}$
10: **for** each h_{ij} in $\mathbb{H}$ **do**
11: Using a pair of binary searches, compute $|h_{ij} \cap S'|$ (similar to the computation of $|O_i|$ in Algorithm 5)
12: $HD_q \leftarrow \min\{HD_q, |h_{ij} \cap S'|\}$
13: $HD(q; S) \leftarrow \frac{2}{n} (HD_q + |Triv|)$
14: **return** $HD(q; S)$

Algorithm 8 Removing-Duplicates(A)

Input: Sorted array A
Output: Sorted array Anc with unique elements

```
1: Define Anc with dynamic size
2: Initialize Anc[1] = A[1], j = 2
3: for i = 2 to |A| do
4:     if (A[i] ≠ A[i − 1])
5:         Anc[j] ← A[i]
6:         j ← j + 1
7: return Anc
```

Chapter 6

Lower Bounds

As discussed in Chapter 3, computing each of simplicial depth and halfspace depth in $\mathbb{R}^2$ requires $\Omega(n \log n)$ time. In this chapter we prove that computing the planar β-skeleton depth also requires $\Omega(n \log n)$, $\beta \geq 1$. We reduce the problem of Element Uniqueness to the problem of computing the β-skeleton depth in three cases $\beta = 1$, $1 < \beta < \infty$, and $\beta = \infty$. It is known that the question of Element Uniqueness has a lower bound of $\Omega(n \log n)$ in the algebraic decision tree model of computation proposed in [19].

6.1 Lower Bound for the Planar β-skeleton Depth, $\beta = 1$

Theorem 6.1.1. *Computing the planar spherical depth (β-skeleton depth, $\beta = 1$) of a query point in the plane takes $\Omega(n \log n)$ time.*

Proof. We show that finding the spherical depth allows us to answer the question of Element Uniqueness. Suppose that $A = \{a_1, ..., a_n\}$, for $n \geq 2$ is a given set of real numbers. We suppose all of the numbers to be positive (negative), otherwise we shift the points onto the positive x-axis. For every $a_i \in A$ we construct four points x_i, x_{n+i}, x_{2n+i}, and x_{3n+i} in the polar coordinate system as follows:

$$x_{(kn+i)} = \left(r_i, \theta_i + \frac{k\pi}{2} \right);\ 0 \leq k \leq 3,$$

where $r_i = \sqrt{1 + a_i^2}$ and $\theta_i = \tan^{-1}(1/a_i)$. Thus we have a set S of $4n$ points x_{kn+i}, for $1 \leq i \leq n$. See Figure 6.1. The Cartesian coordinates of the points can be computed by:

$$x_{(kn+i)} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}^k \begin{pmatrix} a_i \\ 1 \end{pmatrix};\ k = 0, 1, 2, 3.$$

We select the query point $q = (0, 0)$, and present an equivalent form of Equation (5.2) for O_j as follows:

$$O_j = \left\{ x_k \in S \mid \angle x_j q x_k \geq \frac{\pi}{2} \right\},\ 1 \leq j \leq 4n, \tag{6.1}$$

We compute $SphD(q; S)$ in order to answer the Element Uniqueness problem. Suppose that every $x_j \in S$ is a unique element. In this case, $|O_j| = 2n + 1$

because, from (6.1), it can be figured out that the expanded O_j is as follows:

$$O_j = \begin{cases} \{x_{n+1}, ..., x_{n+j}, x_{2n+1}, ..., x_{3n}, x_{3n+j}, ..., x_{4n}\}; & j \in \{1, ..., n\} \\ \{x_{2n+1}, ..., x_{n+j}, x_{3n+1}, ..., x_{4n}, x_{j-n}, ..., x_n\}; & j \in \{n+1, ..., 2n\} \\ \{x_{3n+1}, ..., x_{n+j}, x_1, ..., x_n, x_{j-n}, ..., x_{2n}\}; & j \in \{2n+1, ..., 3n\} \\ \{x_1, ..., x_{j-3n}, x_{n+1}, ..., x_{2n}, x_{j-n}, ..., x_{3n}\}; & j \in \{3n+1, ..., 4n\}. \end{cases}$$

Let $SphD_S(q)$ be the unnormalized form of $SphD(q; S)$. Referring to Theorem 5.1.1 and Equation (5.3),

$$SphD_S(q) = \binom{|S|}{2} SphD(q; S) = \frac{1}{2} \sum_{1 \le j \le 4n} (2n+1) = 4n^2 + 2n.$$

Now suppose that there exist some $i \neq j$ such that $x_i = x_j$ in S. In this case, from Equation (6.1), it can be seen that:

$$|O_{(kn+i) \bmod 4n}| = |O_{(kn+j) \bmod 4n}| = 2n + 2,$$

where $k = 0, 1, 2, 3$ (see Figure 6.1). As an example, for $k = 0$, $|O_j| = |O_i| = 2n + 2$ because the expanded form of these two sets is as follows: (without loss of generality, assume $i < j < n$)

$$O_i = O_j = \{x_{n+1}..x_{n+j}, x_{2n+1}..x_{3n}, x_{3n+i}, x_{3n+j}, x_{3n+j+1}..x_{4n}\}.$$

Theorem 5.1.1 and Equation (5.3) imply that:

$$SphD_S(q) \geq \frac{1}{2}(8 + \sum_{1 \leq j \leq 4n} (2n+1)) = 4n^2 + 2n + 4.$$

Therefore the elements of A are unique if and only if the spherical depth of $(0,0)$ with respect to S is $4n^2 + 2n$. This implies that the computation of spherical depth requires $\Omega(n \log n)$ time. It is necessary to mention that the only computation in the reduction is the construction of S which takes $O(n)$ time. Finally, we mention that the reduction does not depend on the sorted order of the elements. □

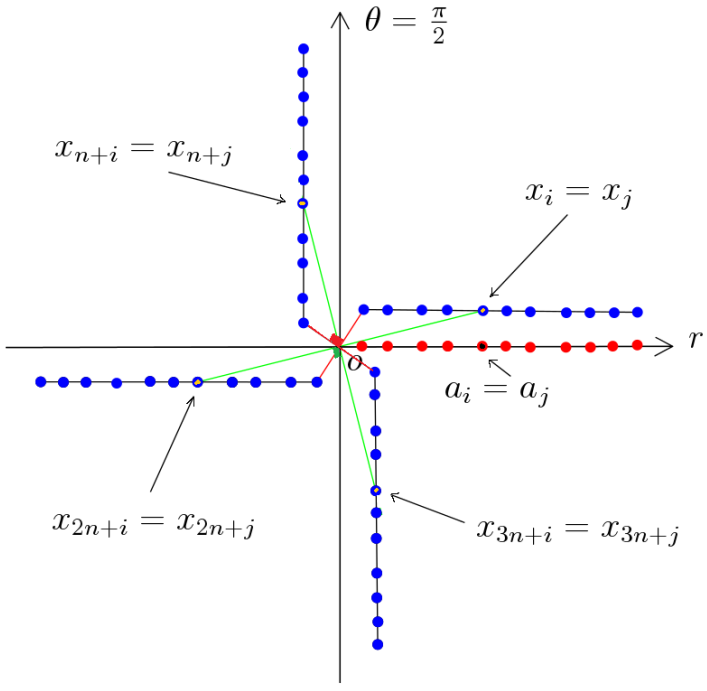

Figure 6.1: An illustration of A and S for $\beta = 1$

Note 6.1.1. *Instead of four copies of the elements of A, we could consider two copies of such elements to construct S as used in section 6.2. However, the depth calculation becomes more complicated in this case.*

6.2 Lower Bound for the Planar β-skeleton Depth, $1 < \beta < \infty$

First, we prove the lower bound for the planar lens depth where $\beta = 2$ in β-skeleton depth. Using the same reduction technique, we generalize the result to all values of $1 < \beta < \infty$.

Lemma 6.2.1. *For $1 \leq i \leq n$ and $b_i \in \mathbb{R}^+$, suppose that S and L_j are two sets of polar coordinates as follows:*

$$S = \{x_i = (b_i, 0), x_{n+i} = (b_i, \pi/3)\}$$

$$L_j = \{x_k \in S \mid O = (0,0) \in L(x_j, x_k)\}\ , 1 \leq j \leq 2n.$$

For a unique element $x_j \in S$, $L_j = \{x_{(n+j) \bmod 2n}\}$.

Proof. Suppose that $x_k \in L_j; k \notin \{j, (n+j) \bmod 2n\}$. We prove that such x_k does not exist. If $\angle x_j O x_k = 0$, it is obvious that $O \notin L(x_j, x_k)$ and x_k cannot be an element of L_j. For the case $\angle x_j O x_k = \pi/3$, by definition of L_j,

$O \in L(x_j, x_k)$. From Definition 3.4.1 for $\beta = 2$,

$$O \in L(x_j, x_k) \Leftrightarrow O \in B(x_j, \|x_j - x_k\|) \cap B(x_k, \|x_j - x_k\|) \tag{6.2}$$

$$\Leftrightarrow \|x_j - O\| \leq \|x_j - x_k\| \;\&\; \|x_k - O\| \leq \|x_j - x_k\| \tag{6.3}$$

$$\Leftrightarrow \|x_j - x_k\| \geq \max\{b_j, b_k\}. \tag{6.4}$$

From the cosine formula[1] in triangle $\triangle x_i O x_j$, we have

$$\|x_j - x_k\|^2 = b_j^2 + b_k^2 - 2b_j b_k \cos(\pi/3). \tag{6.5}$$

Equations (6.4) and (6.5) imply that:

$$b_j^2 + b_k^2 - b_j b_k \geq \max\{b_j^2, b_k^2\} \Leftrightarrow \begin{cases} b_j^2 + b_k^2 - b_j b_k \geq b_j^2 \\ b_j^2 + b_k^2 - b_j b_k \geq b_k^2 \end{cases}$$

$$\Leftrightarrow \begin{cases} b_k^2 - b_j b_k \geq 0 \\ b_j^2 - b_j b_k \geq 0 \end{cases} \Leftrightarrow \begin{cases} b_k \geq b_j \\ b_j \geq b_k \end{cases} \Leftrightarrow b_k = b_j.$$

This result contradicts the assumption of $x_k \in L_j; k \notin \{j, (n+j) \bmod 2n\}$ $\square$

Theorem 6.2.1. *Computing the lens depth of a query point in the plane takes* $\Omega(n \log n)$ *time.*

[1]Cosine formula: For a triangle $\triangle abc$,

$$\|ab\|^2 = \|ac\|^2 + \|bc\|^2 - 2\|ac\|\|bc\| \cos(\angle bca).$$

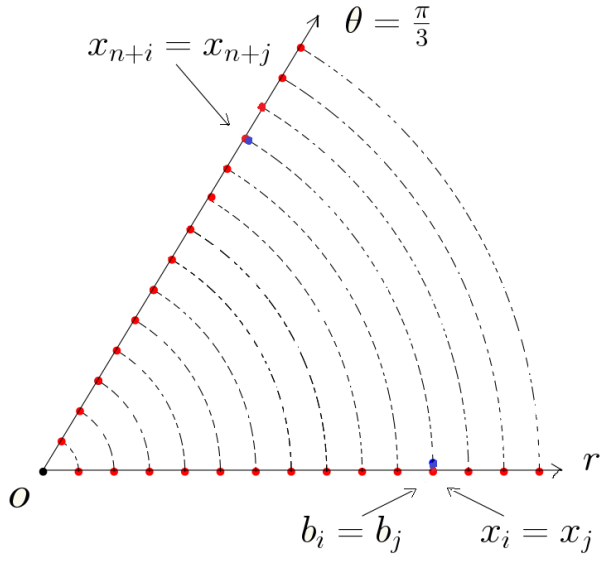

Figure 6.2: An illustration of B and S for $\beta = 2$

Proof. Suppose that $B = \{b_1, ..., b_n\}$, for $n \geq 2$ is a given set of real numbers. Without loss of generality, we let these numbers to be positive (see the proof of Theorem 6.1.1). For $1 \leq i \leq n$, we construct set $S = \{x_i, x_{n+i}\}$ of $2n$ points in the polar coordinate system such that $x_i = (b_i, 0)$ and $x_{n+i} = (b_i, \pi/3)$. See Figure 6.2. We select the query point $q = (0, 0)$, and define L_j as follows:

$$L_j = \{x_k \in S \mid q \in L(x_j, x_k)\}\,,\ 1 \leq j \leq 2n. \tag{6.6}$$

Using Equation (6.6), the unnormalized form of Equation (3.6) can be presented by:

$$LD_S(q) = \binom{|S|}{2} LD(q; S) = \frac{1}{2} \sum_{1 \leq j \leq 2n} |L_j|. \tag{6.7}$$

We solve the problem of Element Uniqueness by computing $LD_S(q)$. Suppose

that every $x_j \in S$ is a unique element. In this case, Lemma 6.2.1 implies that $L_j = \{x_{(n+j) \bmod 2n}\}$. From Equation (6.7), we have

$$LD_S(q) = \frac{1}{2} \sum_{1 \leq j \leq 2n} 1 = n.$$

Now assume that there exists some $i \neq j$ such that $x_i = x_j$ in S. In this case,

$$L_j = L_i = \{x_{(n+i) \bmod 2n}, x_{(n+j) \bmod 2n}\}$$
$$L_{(n+i) \bmod 2n} = L_{(n+j) \bmod 2n} = \{x_i, x_j\}.$$

As such, for $1 \leq t \leq 2n$ and $t \notin \{i, j, (i+n) \bmod 2n, (j+n) \bmod 2n\}$,

$$\begin{aligned} LD_S(q) &= \frac{1}{2} \sum_t |L_t| + \frac{1}{2}(|L_i| + |L_j| + |L_{(i+n) \bmod 2n}| + |L_{(j+n) \bmod 2n}|) \\ &= \frac{1}{2}(2(n-2)) + \frac{1}{2}(2+2+2+2) = n+2. \end{aligned}$$

For the case of having more duplicated elements in S,

$$LD_S(q) = n + 2c, \tag{6.8}$$

where c is the number of duplicates. Therefore the elements of S are unique if and only if $c = 0$ in Equation (6.8). This implies that the computation of lens depth requires $\Omega(n \log n)$ time. Note that all of the other computations in this reduction take $O(n)$. □

Lemma 6.2.2. *For $a, b \in \mathbb{R}^2$, suppose that u_β is a fixed intersection point between the two disks constructing the $S_\beta(a, b)$. $\theta = \angle a u_\beta b = \cos^{-1}(1-1/\beta)$.*

Proof. From Definition 3.4.1,

$$S_\beta(a, b) = B(c_a, (\beta/2)\|a - b\|) \cap B(c_b, (\beta/2)\|a - b\|),$$

where $c_a = (\beta/2)a + (1 - \beta/2)b$ and $c_b = (\beta/2)b + (1 - \beta/2)a$. It can be verified that $\|c_a - c_b\| = (\beta - 1)\|a - b\|$. Suppose that m is the middle point of $\overline{ab}$ and $h = \|u_\beta - m\|$. See Figure 6.3. The value of θ can be computed as follows.

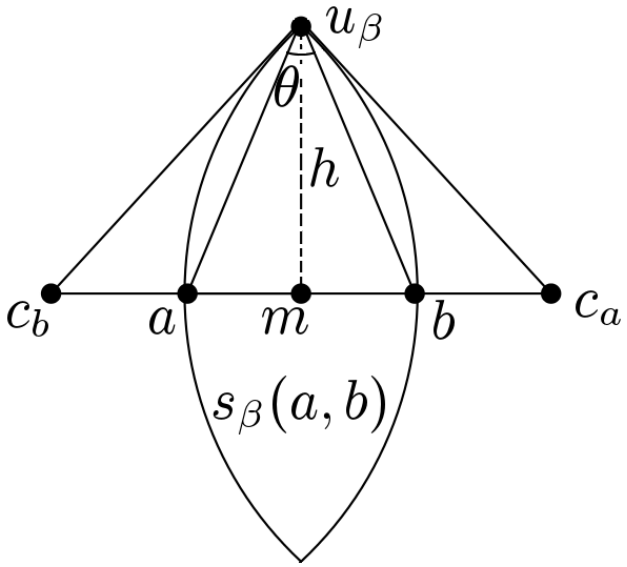

Figure 6.3: The β-influence region for $a, b \in \mathbb{R}^2$.

$$
\begin{aligned}
h^2 = \|u_\beta - a\|^2 - \|a - m\|^2 &= \|u_\beta - c_b\|^2 - \|c_b - m\|^2 \\
&= (\frac{\beta}{2}\|a - b\|)^2 - (\frac{\beta - 1}{2}\|a - b\|)^2 \\
&= \frac{2\beta - 1}{4}\|a - b\|^2 \\
\Rightarrow \|u_\beta - a\|^2 = \frac{2\beta - 1}{4}\|a - b\|^2 + \|a - m\|^2 &= \frac{\beta}{2}\|a - b\|^2
\end{aligned}
$$

The cosine formula in triangle $\triangle a u_\beta b$ implies that

$$
\begin{aligned}
\|a - b\|^2 &= \|u_\beta - a\|^2 + \|u_\beta - b\|^2 - 2\|u_\beta - a\|\|u_\beta - b\|\cos(\theta) \\
&= 2\|u_\beta - a\|^2 - 2\|u_\beta - a\|^2\cos(\theta) \\
&= \beta\|a - b\|^2 - \beta\|a - b\|^2\cos(\theta) \Rightarrow \theta = \cos^{-1}(1 - \frac{1}{\beta}).
\end{aligned}
$$

□

Theorem 6.2.2. *For $1 < \beta < \infty$, computing the β-skeleton depth of a query point in the plane requires $\Omega(n \log n)$ time.*

Proof sketch. *It is enough to generalize the reduction technique in Theorem 6.2.1. As Lemma 6.2.2 suggests, we need to choose $\theta = cos^{-1}(1 - 1/\beta)$ to construct $S = \{x_i = (b_i, 0), x_{n+i} = (b_i, \theta)\}$, where $1 \leq i \leq n$ and $b_i \in B$ defined in the proof of Theorem 6.2.1. Figure 6.4 illustrates that for every unique element $x_i \in S$, there exists only one element in S such that the corresponding β-influence region contains O. As can be seen in this figure, O is not contained in the β-influence region $S_\beta(x_i, y)$. Similar to the proof of Theorem 6.2.1, it can be deduced that $SkD_\beta(O; S) = n$ if every element in*

S is unique. However, $SkD_\beta(O;S) = n+2c$ if there exist c duplicates among the elements of S. Note that we use the real RAM model of computation in order to calculate θ, where we need the square root of a real number to be computed in constant time.

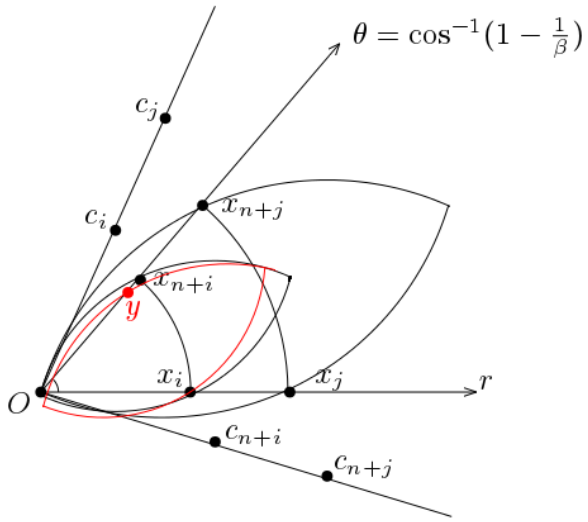

Figure 6.4: The β-influence regions for some elements of S when $1 < \beta < \infty$.

6.3 Lower Bound for the Planar β-skeleton Depth, $\beta = \infty$

Suppose that B is a set of positive real numbers as introduced in the proof of Theorem 6.2.1. From the proof of Theorem 6.2.2, the rotation angle $\theta = cos^{-1}(1 - 1/\beta)$ is equal to 0 if $\beta = \infty$. It means that there is not a proper rotation angle to make the second copy of the data points. However, it is

enough to shift up the points by some constant (e.g. $\max\{b_i\}$), and construct $S = \{x_i = (b_i, 0), x_{n+i} = (b_i, \max\{b_i\})\}$ (see Figure 6.5).

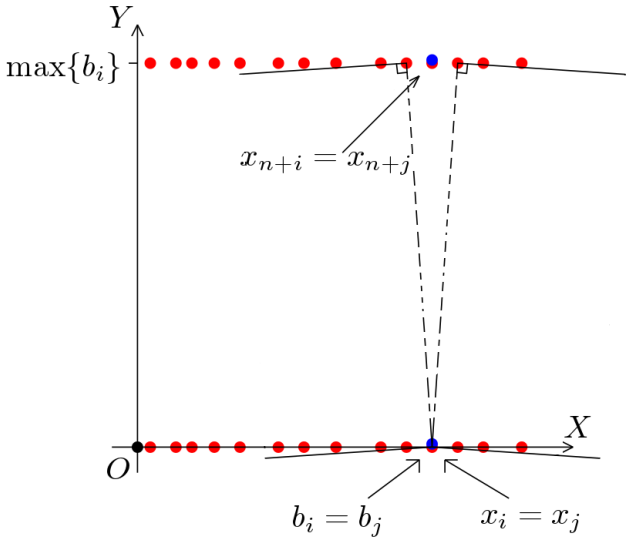

Figure 6.5: An illustration of B and S when $\beta = \infty$

We select $q = (0, 0)$, and define

$$S_{\infty j} = \{x_k \in S \mid q \in S_\infty(x_j, x_k)\}\,,\ 1 \leq j \leq 2n. \tag{6.9}$$

From Definition 3.4.1, it can be verified that

$$q \in S_\infty(x_j, x_k) \Leftrightarrow \max\{\angle q x_k x_j, \angle q x_j x_k\} \leq \frac{\pi}{2}.$$

Suppose that every $x_i \in S; 1 \leq i \leq n$ is a unique element. In this case,

$$S_{\infty i} = \{x_{n+1}, x_{n+2}, ..., x_{n+i}\},\ S_{\infty(n+i)} = \{x_i, x_{i+1}, ..., x_n\}. \tag{6.10}$$

From Equation (6.9), the unnormalized form of Equation (3.4) for $\beta = \infty$ can be presented by:

$$SkD_{\infty S}(q) = \binom{|S|}{2} SkD_{\infty}(q; S) = \frac{1}{2} \sum_{1 \leq j \leq 2n} |S_{\infty j}|. \tag{6.11}$$

Equations (6.10) and (6.11) imply that

$$\begin{aligned} SkD_{\infty S}(q) &= \frac{1}{2} \sum_{1 \leq i \leq n} (|S_{\infty i}| + |S_{\infty (n+i)}|) \\ &= \frac{1}{2} \sum_{1 \leq i \leq n} (i + (n - i + 1)) = \frac{n(n+1)}{2} = \binom{n+1}{2}. \end{aligned}$$

Now assume that for some $i \neq k$, $x_i = x_k$ in S. Without loss of generality, suppose that $1 \leq i < k \leq n$. In this case,

$$S_{\infty i} = S_{\infty k} = \{x_{n+1}, x_{n+2}, ..., x_{n+i}, x_{n+k}\} \tag{6.12}$$

$$S_{\infty (n+i)} = S_{\infty (n+k)} = \{x_k, x_i, x_{i+1}, ..., x_n\}. \tag{6.13}$$

As such, for $1 \leq t \leq n$ and $t \notin \{i, k\}$, Equations (6.10), (6.11), (6.12), and (6.13) can be used to compute $SkD_{\infty S}(q)$ as follows.

$$
\begin{aligned}
SkD_{\infty S}(q) &= \frac{1}{2} \sum_{1 \le j \le 2n} |S_{\infty j}| \\
&= \frac{1}{2} \left(\sum_t (|S_{\infty t}| + S_{\infty (n+t)}|) + |S_{\infty i}| + |S_{\infty (n+i)}| + |S_{\infty k}| + |S_{\infty (n+k)}| \right) \\
&= \frac{1}{2} \left(\sum_t (t + n - t + 1) + (i + 1) + (n - i + 2) + (k + 1) + (n - k + 2) \right) \\
&= \frac{1}{2} \left(\sum_t (n + 1) + 2(n + 3) \right) = \frac{1}{2} \left((n - 2)(n + 1) + 2(n + 1) + 4 \right) \\
&= \frac{1}{2} (n(n + 1) + 4) = \binom{n+1}{2} + 2
\end{aligned}
$$

For the general case of having c duplicates among the elements of S,

$$
SkD_{\infty S}(q) = \binom{n+1}{2} + 2c. \tag{6.14}
$$

Therefore the elements of S are unique if and only if $c = 0$ in Equation (6.14). The above results imply that the computation of β-skeleton depth, where $\beta = \infty$, requires $\Omega(n \log n)$ time.

Chapter 7

Relationships and Experiments

In this chapter we study the relationships among different depth functions such as β-skeleton depth, halfspace depth, and simplicial depth in two different ways. First, we focus on the geometric properties of the influence regions. Second, the idea of fitting function is applied to approximate one data depth using another one. Our main motivation to study the relationships among different depth functions is derived from the complexity of computations, especially in higher dimensions. For example, computing the β-skeleton depth using brute force algorithm is much easier and relatively faster than computing most of the other depth functions such as halfspace depth and simplicial depth. Unlike halfspace depth and simplicial depth, the time complexity of β-skeleton depth grows linearly in the dimension. Recall that the time complexity of β-skeleton depth using a brute force algorithm in dimension d is $O(dn^2)$. Whereas, the best known algorithm for computing the simplicial

depth in the higher dimension d is brute force which takes $O(n^{d+1})$ time. Computing the halfspace depth is an NP-hard problem when the dimension d is a part of input. See Sections 3.2, 3.3, and 3.4.

7.1 Geometric Relationships

Some geometric properties related to the β-influence regions and simplices are explored in this section. These properties help to bound each one of β-skeleton depth and simplicial depth in terms of the other one.

7.1.1 Convergence of β-skeleton Depth

Lemma 7.1.1. *For $\beta' > \beta \geq 1$ and $a, b \in \mathbb{R}^2$, $S_\beta(a,b) \subseteq S_{\beta'}(a,b)$, where the β-influence region $S_\beta(a,b)$ is the intersection of two disks $B(C_{ab\beta}, R_{ab\beta})$ and $B(C_{ba\beta}, R_{ba\beta})$, $C_{ab\beta} = (\beta/2)(a-b)+b$, and $R_{ab\beta} = (\beta/2)d(a,b)$.*

Proof. To prove $S_\beta(a,b) \subseteq S_{\beta'}(a,b)$ which is equivalent with Equation (7.1),

$$B(C_{ab\beta}, R_{ab\beta}) \cap B(C_{ba\beta}, R_{ba\beta}) \subseteq B(C_{ab\beta'}, R_{ab\beta'}) \cap B(C_{ba\beta'}, R_{ba\beta'}) \qquad (7.1)$$

it suffices to prove both inclusion relationships $B(C_{ab\beta}, R_{ab\beta}) \subseteq B(C_{ab\beta'}, R_{ab\beta'})$ and $B(C_{ba\beta}, R_{ba\beta}) \subseteq B(C_{ba\beta'}, R_{ba\beta'})$. We only prove the first one, and the second one can be proved similarly. Suppose that $\beta < \beta' = \beta + \varepsilon; \varepsilon > 0$. It is trivial to check that two disks $B(C_{ab\beta}, R_{ab\beta})$ and $B(C_{ab\beta'}, R_{ab\beta'})$ meet at b. See Figure 7.1. Let $t \neq b$ be an extreme point of $B(C_{ab\beta}, R_{ab\beta})$. This implies

that

$$\begin{aligned} d(t, C_{ab\beta}) = R_{ab\beta} &\Leftrightarrow d(t, \frac{\beta(a-b)}{2} + b) = \frac{\beta(d(a,b))}{2} \\ &\Leftrightarrow \left\| \frac{\beta(a-b)}{2} + (b-t) \right\|^2 = \left(\frac{\beta(a-b)}{2} \right)^2 \\ &\Leftrightarrow \|b-t\|^2 - \beta(b-a) \cdot (b-t) = 0. \end{aligned}$$

The last equality means that $(b-a) \cdot (b-t) \geq 0$. Hence,

$$\begin{aligned} &\|b-t\|^2 - \beta(b-a) \cdot (b-t) = 0 \\ &\Leftrightarrow \|b-t\|^2 - (\beta+\varepsilon)(b-a) \cdot (b-t) < 0 \\ &\Leftrightarrow \|b-t\|^2 - \beta'(b-a) \cdot (b-t) < 0 \\ &\Leftrightarrow d(t, C_{ab\beta'}) - R_{ab\beta'} < 0 \\ &\Leftrightarrow t \in intB(C_{ab\beta'}, R_{ab\beta'}). \end{aligned}$$

From the above calculations, every extreme point of $B(C_{ab\beta}, R_{ab\beta})$ is an interior point of $B(C_{ab\beta'}, R_{ab\beta'})$; therefore, $B(C_{ab\beta}, R_{ab\beta}) \subseteq B(C_{ab\beta'}, R_{ab\beta'})$. □

Lemma 7.1.2. *Suppose that $\beta' > \beta \geq 1$. For a query point q and given data set S in $\mathbb{R}^2$, $SkD_{\beta'}(q; S) \geq SkD_{\beta}(q; S)$.*

Proof. For any arbitrary pair of points x_i and x_j in S, Lemma 7.1.1 implies that $S_\beta(x_i, x_j) \subseteq S_{\beta'}(x_i, x_j)$. Hence, Equation (7.2) is sufficient to complete

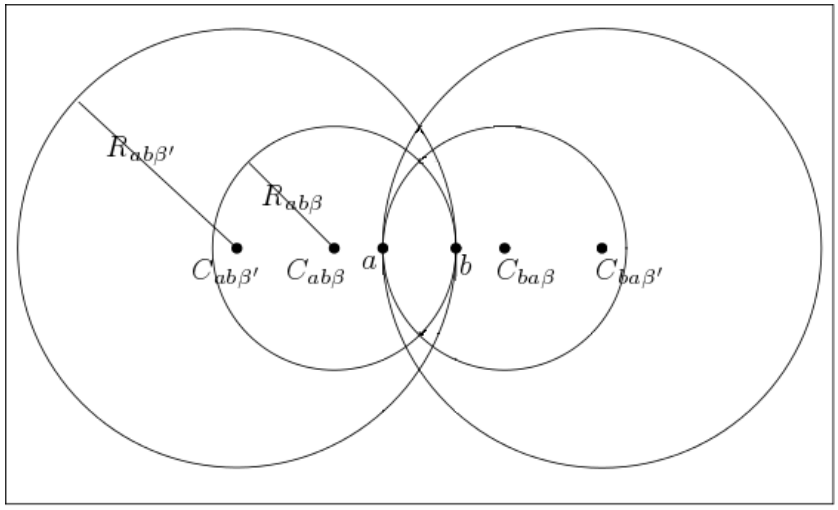

Figure 7.1: $S_\beta(a,b)$ and $S_{\beta'}(a,b)$.

the proof.

$$SkD_{\beta'}(q;S) = \sum_{x_i,x_j \in S} I(q \in S_{\beta'}(x_i,x_j)) \geq \sum_{x_i,x_j \in S} I(q \in S_\beta(x_i,x_j)) = SkD_\beta(q;S) \tag{7.2}$$

□

Definition 7.1.1. *A query point q is generic with respect to a data set $S = \{x_1, x_2, \ldots, x_n\}$ if for all $x_i, x_j \in S$, q does not lie on the boundary of $S_\infty(x_i, x_j)$ or, on the line segment $\overline{x_i x_j}$.*

Lemma 7.1.3. *Suppose that $S \subset \mathbb{R}^2$ is a given data set and $Q \subset \mathbb{R}^2$ is a set of generic query points. Assuming that $\mathcal{R} \subset \mathbb{R}^2$ is a large enough finite range that contains S and Q, for two distinct elements x_i and x_j in S, and $q \in Q$,*

$$\exists \beta_{ij} < \infty; q \in S_\infty(x_i,x_j) \Rightarrow q \in S_{\beta_{ij}}(x_i,x_j). \tag{7.3}$$

There also exists a $\beta^* < \infty$ that satisfies (7.3) *for all $\{x_i, x_j\} \subset S, x_i \neq x_j$.*

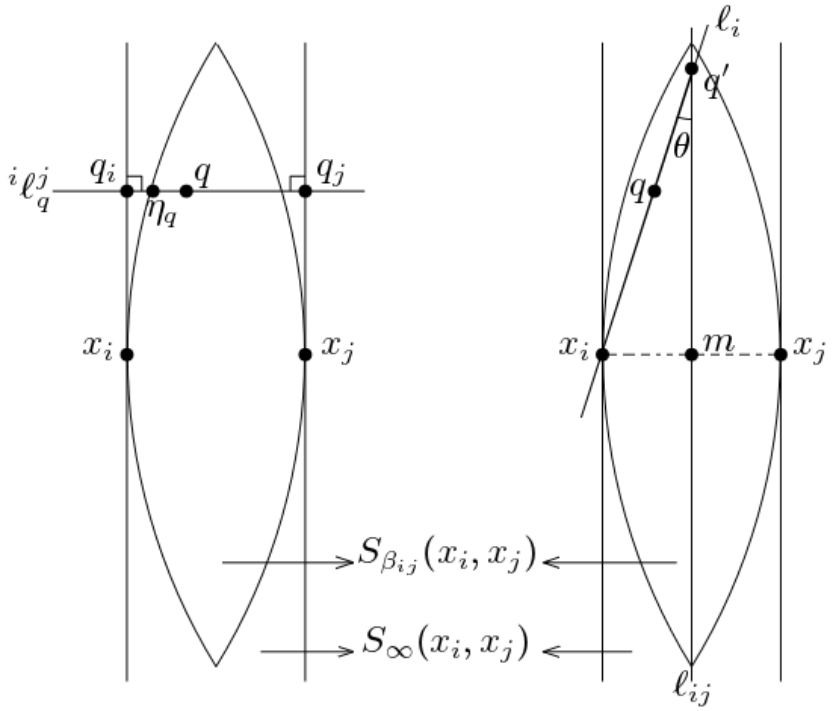

Figure 7.2: Slab and β_{ij}-influence region containing point q

Proof. Suppose that q is a generic query point, m is the middle point of $\overline{x_i x_j}$, and ℓ_{ij} is a line parallel to the boundaries of $S_{\beta_{ij}}(x_i, x_j)$, through m. We consider two lines ℓi, through q and x_i, and ℓ_j, through q and x_j. Obviously, each of ℓ_i and ℓ_j intersects ℓ_{ij}. Between these two intersections, let q' be the farthest one. See Figure 7.2 (right). The desired value of β_{ij} in Equation (7.3) can be computed using Lemma 6.2.2 as follows.

$$\theta = \cos^{-1}(1 - \frac{1}{\beta_{ij}}) \Rightarrow \cos(\theta) = 1 - \frac{1}{\beta_{ij}} \Rightarrow \beta_{ij} = \frac{1}{1 - \cos(\theta)} \tag{7.4}$$

where θ is the smaller angle at q'. Since q is a generic query point, $\cos(\theta) \neq 1$. Equation (7.4) provides the desired value for β_{ij} in Equation (7.3). The proof is complete because

$$\forall\{x_i, x_j\} \subset S; q \in S_\infty(x_i, x_j) \Rightarrow q \in S_{\beta^*}(x_i, x_j),\ \beta^* = \max_{ij}\{\beta_{ij}\}. \tag{7.5}$$

Another proof: Suppose that q is a generic query point, and $q \in S_\infty(x_i, x_j)$. Let ${}^i\ell_q^j$ be a line passing through q, and perpendicular to the boundaries of slab $S_\infty(x_i, x_j)$ in points q_i and q_j. See Figure 7.2 (left). Every value of β_{ij} that meets the following requirements can be considered as the desired β_{ij}.

- β_{ij} is large enough such that $S_{\beta_{ij}}(x_i, x_j)$ is cut by ${}^i\ell_q^j$.
- $\min\{\|q - q_i\|, \|q - q_j\|\} \geq \min\{\|q_i - \eta_q\|, \|q_j - \eta_q\|\}$, where η_q is a fixed intersection point between ${}^i\ell_q^j$ and $S_{\beta_{ij}}(x_i, x_j)$.

The value of β^* can be chosen as in Equation (7.5). □

Theorem 7.1.1. *For data set S and a generic query point q given in a large enough finite range $\mathcal{R} \subset \mathbb{R}^2$, the β-skeleton depth functions converge. In other words,*

$$\exists\beta^* < \infty; \forall\beta \geq \beta^*,\ SkD_\beta(q; S) = S_\infty(q; S). \tag{7.6}$$

Proof. It is enough to prove that there exists a $\beta^* < \infty$ such that

$$\forall\{x_i, x_j\} \subset S, x_i \neq x_j, q \in S_{\beta^*}(x_i, x_j) \Leftrightarrow q \in S_\infty(x_i, x_j). \tag{7.7}$$

$\Rightarrow$) Lemma 7.1.1 is obviously enough because

$$\forall\{x_i, x_j\} \subset S, \forall\beta^* < \infty, S_{\beta^*}(x_i, x_j) \subset S_\infty(x_i, x_j).$$

$\Leftarrow$) To prove this direction, Lemma 7.1.3 suggests that it is enough to choose

$$\beta^* = \max_{ij}\{\beta_{ij}\}.$$

□

7.1.2 β-skeleton Depth versus Simplicial Depth

First, we study the relationship between spherical (β-skeleton, $\beta = 1$) depth and simplicial depth. From Lemma 7.1.2, we can generalize the obtained results for every value of β.

Definition 7.1.2. *For a point $q \in \mathbb{R}^2$ and a data set $S = \{x_1, ..., x_n\} \subset \mathbb{R}^2$, we define $B_{in}(q; S)$ to be the set of all closed spherical influence regions, out of $\binom{n}{2}$ possible of them, that contain q. We also define $S_{in}(q; S)$ to be the set of all closed triangles, out of $\binom{n}{3}$ possible defined by S, that contain q.*

Lemma 7.1.4. *Suppose that q is a point inside $Conv(S)$, where $S \subset \mathbb{R}^2$ is a given data set. q is covered by the union of spherical influence regions defined by S.*

Proof. Let $H = Conv(S)$. By Caratheodory's theorem [24], there is at least one triangle, defined by the vertices of H, that contains q. We prove that

the union of the spherical influence regions defined by such triangle contains q. See Figure 7.3. This statement can be proved by contradiction. Suppose that q is covered by none of $Sph(a,b)$, $Sph(a,c)$, and $Sph(b,c)$. Therefore, Theorem 5.1.1 implies that none of the angles $\angle aqb$, $\angle aqc$, and $\angle bqc$ is greater than or equal to $\frac{\pi}{2}$ which is a contradiction because at least one of these angles should be at least $\frac{2\pi}{3}$ in order to get 2π as their sum. □

Lemma 7.1.5. *Suppose that $S = \{a, b, c\}$ is a set of points in $\mathbb{R}^2$. For every $q \in \mathbb{R}^2$, if $|S_{in}(q; S)| = 1$, then $|B_{in}(q; S)| \geq 2$.*
Another form of Lemma 7.1.5 is that if $q \in \triangle abc$, then q falls inside at least two spherical influence regions out of $Sph(a,b)$, $Sph(c,b)$, and $Sph(a,c)$. The equivalency between these two forms of the lemma is clear. We prove the first one.

Proof. From Lemma 7.1.4, $|B_{in}(q; S)| \geq 1$. Suppose that $|B_{in}(q; S)| = 1$. If q is one of the vertices of $\triangle abc$, it is clear that $|B_{in}(q; S)| \geq 2$. Without loss of generality, we suppose that q falls in $int\ Sph(a,b)$. For the rest of the proof, we focus on the relationships among the angles $\angle aqb$, $\angle cqa$, and $\angle cqb$ (see Figure 7.3). Since q is inside $\triangle abc$, $\angle aqb \leq \pi$. Consequently, at least one of $\angle cqa$ and $\angle cqb$ is greater than or equal to $\pi/2$. So, Theorem 5.1.1 implies that q is in at least one of $int\ Sph(a,c)$ and $int\ Sph(b,c)$. Hence, $|B_{in}(q; S)| = 1$ contradicts $|S_{in}(q; S)| = 1$ which means that $|B_{in}(q; S)| \geq 2$. As an illustration, in Figure 7.3, for the points in the hatched area $|B_{in}(q; S)| = 3$. □

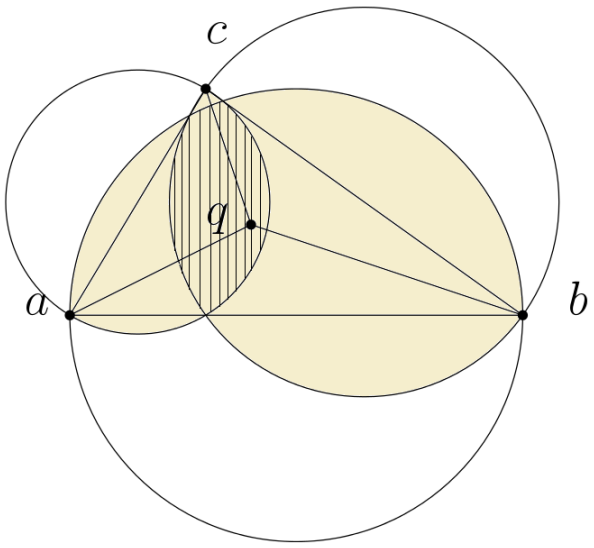

Figure 7.3: Triangle *abc* and its corresponding spherical influence regions containing q

Lemma 7.1.6. *For a data set* $S = \{x_1, ..., x_n\} \subset \mathbb{R}^2$,

$$\frac{|B_{in}(q;S)|}{|S_{in}(q;S)|} \geq \frac{2}{n-2}.$$

Proof. Suppose that $Sph(x_i, x_j) \in B_{in}(q;S)$. There exist at most $(n-2)$ triangles in $S_{in}(q;S)$ such that $\overline{x_i x_j}$ is an edge of them. We consider $\triangle x_i x_j x_k$ to be one of such triangles (see Figure 7.4 as an illustration). Referring to Lemma 7.1.5, q belongs to at least one of $Sph(x_i, x_k)$ and $Sph(x_j, x_k)$. Similarly, there exist at most $(n-2)$ triangles in $S_{in}(q;S)$ such that $x_i x_k$ (respectively $x_j x_k$) is an edge of them. In the process of computing $|S_{in}(q;S)|$, triangle $\triangle x_i x_j x_k$ is counted at least two times, once for $Sph(x_i, x_j)$ and an-

other time for $Sph(x_i, x_k)$ (or $Sph(x_j, x_k)$). Consequently, for every sphere area in $B_{in}(q; S)$, there exist at most $\frac{(n-2)}{2}$ distinct triangles, triangles with only one common side, in $S_{in}(q; S)$. As a result, Equation (7.8) can be obtained.

$$\frac{(n-2)}{2}|B_{in}(q;S)| \geq |S_{in}(q;S)| \Rightarrow \frac{|B_{in}(q;S)|}{|S_{in}(q;S)|} \geq \frac{2}{(n-2)} \tag{7.8}$$

□

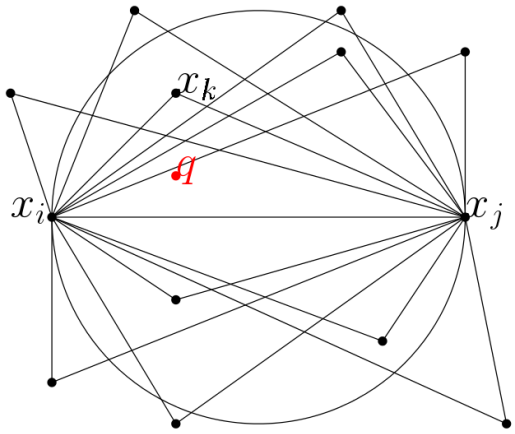

Figure 7.4: $Sph(x_i, x_j)$ and all triangles, with edge $x_j x_j$, containing q

Theorem 7.1.2. *For a data set* $S = \{x_1, ..., x_n\}$ *and a query point* q *in* $\mathbb{R}^2$, $SphD(q; S) \geq \frac{2}{3} SD(q; S)$.

Proof. From Definition 7.1.2 the definitions of spherical depth and simplicial

depth, (7.9) and (7.10) can be easily verified.

$$SphD(q;S) = \frac{1}{\binom{n}{2}}|B_{in}(q;S)| \tag{7.9}$$

$$SD(q;S) = \frac{1}{\binom{n}{3}}|S_{in}(q;S)| \tag{7.10}$$

Using these two equations, the ratio of spherical depth and simplicial depth can be calculated as follows:

$$\frac{SphD(q;S)}{SD(q;S)} = \frac{\binom{n}{3}}{\binom{n}{2}}\frac{|B_{in}(q;S)|}{|S_{in}(q;S)|} = \frac{(n-2)|B_{in}(q;S)|}{3|S_{in}(q;S)|}. \tag{7.11}$$

Equation (7.11) and Lemma 7.1.6 imply that

$$\frac{SphD(q;S)}{SD(q;S)} \geq \frac{2(n-2)}{3(n-2)} = \frac{2}{3} \Rightarrow SphD(q;S) \geq \frac{2}{3}\,SD(q;S).$$

□

Theorem 7.1.3. *Suppose that S is a given data set consisting of n points in general position in $\mathbb{R}^2$. For $q \in \mathbb{R}^2$ and $\beta \geq 1$, $SkD_\beta(q;S) \geq \frac{2}{3}\,SD(q;S)$.*

Proof. The proof can be derived from Lemma 7.1.2 and Theorem 7.1.2. □

7.2 Relationships via Dissimilarity Measures

In this section we define two dissimilarity measures between every pair of depth functions, and approximate these depth functions by one another with

a certain amount of error. A notable application of this approximation technique is that we can approximate the halfspace depth, which is an NP-hard problem in higher dimension d, by the β-skeleton depth which takes only $O(dn^2)$ time (see Section 7.3). The dissimilarity measures in this technique are defined based on the concepts of *fitting function* and *Hamming distance*. We train the halfspace depth function by the β-skeleton depth values obtaining from a given data set. The goodness of approximation can be determined using the dissimilarity measures and the sum of squares of error values.

7.2.1 Fitting Function and Dissimilarity Measure

To determine the dissimilarity between two vectors $U = (u_1, ..., u_n)$ and $V = (v_1, ..., v_n)$, the idea of fitting functions can be applied. Considering the goodness measures of fitting functions in Section 2.2.3, assume that f is the best function fitted to U and V. In other words, $u_i = f(v_i) \pm \delta_i$; $\delta_i \geq 0$ and $1 \leq i \leq n$. Let $\xi_i = u_i - \overline{U}$, where $\overline{U}$ is the average of u_i $(1 \leq i \leq n)$. In Equation (7.12), we define $d_E(U, V)$, as a function of δ_i and ξ_i, to measure the dissimilarity between U and V.

$$d_E(U, V) = 1 - r^2, \tag{7.12}$$

where r^2 is the coefficient of determination. We recall from Equation (2.1) that

$$r^2 = \frac{\sum_{i=1}^{n}(\xi_i^2 - \delta_i^2)}{\sum_{i=1}^{n}\xi_i^2}.$$

Since $r^2 \in [0,1]$, $d_E(U,V) \in [0,1]$. A smaller value of $d_E(U,V)$ represents more similarity between U and V.

7.2.2 Dissimilarity Measure Between two Posets

The idea of defining the following distance comes from the proposed structural dissimilarity measure between posets in [39]. Let $\mathbb{P} = \{P_t = (S, \preceq_t) | t \in \mathbb{N}\}$ be a finite set of posets, where $S = \{x_1, ..., x_n\}$ is a given data set. For $P_k \in \mathbb{P}$ we define a matrix $M^k_{n\times n}$ by:

$$M^k_{ij} = \begin{cases} 1 & x_i \preceq_k x_j \\ 0 & \text{otherwise.} \end{cases}$$

We use the notation of $d_c(P_f, P_g)$ to define the dissimilarity between two posets $P_f, P_g \in \mathbb{P}$ as follows:

$$d_c(P_f, P_g) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}|M^f_{ij} - M^g_{ij}|}{n^2 - n} \tag{7.13}$$

It can be verified that $d_c(P_f, P_g) \in [0, 1]$, where the closer value to 1 means the less similarity between P_f and P_g. This measure of similarity is a metric on $\mathbb{P}$ because for all $P_f, P_g, P_h \in \mathbb{P}$,

- $d_c(P_f, P_g) \geq 0$
- $d_c(P_f, P_g) = 0 \Leftrightarrow P_f = P_g$
- $d_c(P_f, P_g) = d_c(P_g, P_f)$
- $d_c(P_f, P_h) \leq d_c(P_f, P_g) + d_c(P_g, P_h)$.

Proving these properties is straightforward. We prove the last property which is less trivial.

$$\begin{aligned} d_c(P_f, P_h) &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} |M_{ij}^f - M_{ij}^h|}{n^2 - n} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} |(M_{ij}^f - M_{ij}^g) + (M_{ij}^g - M_{ij}^h)|}{n^2 - n} \\ &\leq \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} (|M_{ij}^f - M_{ij}^g| + |M_{ij}^g - M_{ij}^h|)}{n^2 - n} \\ &= \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} |M_{ij}^f - M_{ij}^g|}{n^2 - n} + \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} |M_{ij}^g - M_{ij}^h|}{n^2 - n} \\ &= d_c(P_f, P_g) + d_c(P_g, P_h). \end{aligned}$$

7.3 Approximation of Halfspace Depth

We use the proposed method in Section 7.2 to approximate the halfspace depth. Motivated by statistical applications and machine learning tech-

niques, we train the halfspace depth function using the values of β-skeleton depth. Among all depth functions, the β-skeleton depth is chosen because it is easy to compute and its time complexity, i.e. $O(dn^2)$, grows linearly with the dimension d.

7.3.1 Approximation of Halfspace Depth and Fitting Function

Suppose that $S = \{x_1, ..., x_n\}$ is a given data set. By choosing some subsets of S as training samples, we consider the problem of learning the halfspace depth function using the β-skeleton depth values. In particular, we use the cross validation and information criterion to obtain the best function f such that $HD(x_i; S) = f(SkD_\beta(x_i; S)) \pm \delta_i$. The function f can be considered as an approximation function for halfspace depth. Finally, $d_E(HD, SkD_\beta)$ as the error of approximation can be computed using Equation (7.12).

7.3.2 Approximation of Halfspace Depth and Poset Dissimilarity

In some applications, the structural ranking among the elements of S is more important than the depth value of single points. Let $S = \{x_1, ..., x_n\}$ be a given data set and D be a depth function. Applying D on x_i with respect to S generates a poset. In fact, $P_D = (D(x_i; S), \leq)$ is a chain because for every $x_i, x_j \in S$, the values of $D(x_i; S)$ and $D(x_j; S)$ are comparable. For halfspace

depth and β-skeleton depth, their dissimilarity measure of rankings can be obtained by Equation (7.13) as follows:

$$d_c(HD, SkD_\beta) = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n} |M_{ij}^{HD} - M_{ij}^{SkD_\beta}|}{n^2 - n}.$$

The smaller value of $d_c(HD, SkD_\beta)$, the more similarity between HD and SkD_β in ordering the elements of S.

In sections 7.3.1 and 7.3.2, instead of β-skeleton, any other depth function can be considered to approximate halfspace depth. Considering any other depth function, we can compute the goodness of approximation using dissimilarity measures d_E and d_c.

7.4 Experimental Results

In this section we provide some experimental results to support Sections 7.1, 7.2, and 7.3. The results are summarized in some tables and graphs presented in Sections 7.3.1 and 7.3.2. To obtain these results, we computed the depth functions and their relationships for three sets Q_1, Q_2, and Q_3 of planar query points with respect to data sets S_1, S_2, and S_3 of planar points, respectively. The cardinalities of Q_i and S_i are as follows: $|Q_1| = 100$, $|S_1| = 750$, $|Q_2| = 1000$, $|S_2| = 2500$, $|Q_3| = 2500$, $|S_3| = 10000$. The elements of Q_i and S_i $(i = 1, 2, 3)$ are some randomly generated points (double precision

floating point) within the square $\{(x, y)|x, y \in [-10, 10]\}$. The following lines of code, in MATLAB, are used to generate the elements of Q_i and S_i.

```
% To Generate the Sets of Random Data Points and  Query Points:
n = 3; % the number of decimal places
d = 2; % dimension of points
kS = 10000; % number of data points
kQ = 2500; % number of query points
% [l_range,u_range] is the interval where the
% random points are generated within
l_range = -10; u_range = 10;
S = randi([l_range,u_range]*10^n,[kS,d])/10^n;
Q = randi([l_range,u_range]*10^n,[kQ,d])/10^n;
```

The implementations are done in Java. The source codes and detailed results are publicly available at `https://github.com/RasoulShahsavari/Data-Depth-Source-Codes`.

7.4.1 Experiments for Geometric Relationships

To support the obtained relationships in Section 7.1, we compute the spherical depth, lens depth, and the simplicial depth of the points in three random sets Q_1, Q_2, and Q_3 with respect to data sets S_1, S_2, and S_3, respectively. The results of our experiments are summarized in Table 7.1. Every cell in

the table represents the corresponding depth of query points in Q_i with respect to data set S_i. As can be seen, there are some gaps between obtained experimental bounds for random points and the theoretical bounds in Theorem 7.1.3 and Lemma 7.1.2. For example, the experiments suggests 1.2 as a lower bound for $LD/SphD$, whereas Theorem 7.1.3 introduces the lower bound 1. More research on this topic is needed to figure out if the real bounds are closer to the experimental bounds or to the current theoretical bounds.

	$(Q_1; S_1)$		$(Q_2; S_2)$		$(Q_3; S_3)$	
	Min	Max	Min	Max	Min	Max
SD	0.00	0.25	0.00	0.25	0.00	0.24
$SphD$	0.01	0.50	0.00	0.50	0.00	0.50
LD	0.05	0.61	0.05	0.61	0.04	0.61
$\frac{SphD}{SD}$	2.00	∞	2.00	∞	2.03	∞
$\frac{LD}{SD}$	2.43	∞	2.44	∞	2.44	∞
$\frac{LD}{SphD}$	1.21	8.11	1.22	23.16	1.22	157.16

Table 7.1: Summary of experiments for the geometric relationships among simplicial depth (SD), spherical depth ($SphD$), and lens depth (LD)

7.4.2 Experiments for Approximations

In this section some experimental results are provided to support Theorem 7.1.1, and our proposed method of approximation in Section 7.3. We compute the planar halfspace depth and planar β-skeleton depth of $q \in Q_2$ with respect to S_2 for different values of β, where Q_2 and S_2 are introduced in Section 7.4. The results of our experiments are summarized in Tables 7.2, 7.3,

and 7.4. For each row of these three tables, a plot labelled by the corresponding fitting function is provided (Figures 7.5 - 7.19). As can be seen in the last two rows of Table 7.2, considering $\beta^* = 1000$, Theorem 7.1.1 is supported by these experiments. From Table 7.3, it can be seen that the halfspace depth can be approximated by a quadratic function of the β-skeleton depth with relatively small values of $d_E(SkD, HD) \approxeq 0.0186$ and $d_C(SkD, HD) \approxeq 0.0373$. In particular, $HD \approxeq 3.8883(SkD_\beta - 0.3004)^2$ if $\beta \to \infty$. The results in Table 7.4 indicate that the halfspace depth function can also be approximated via a power model. In particular, $HD \approxeq (SkD_\beta + 0.2255)^{5.8850}$ if $\beta \to \infty$. Referring to the results in Tables 7.3 and 7.4 and Figures 7.10 - 7.19, the power model suggests a better fit both theoretically and visually. For example, the values of d_E in the Table 7.4 are smaller than the corresponding values in Table 7.3. Furthermore, from Figures 7.14 and 7.19, it can be seen that the power model captures the curvature of the data better than the quadratic model does. We recall that d_C represents the structural dissimilarity between the exact values of two depth functions; hence, the values of d_C are not affected by the choice of fitting model. In short, our experimental results on approximating the halfspace depth by β-skeleton depth show that the power model behaves slightly better than the quadratic model. However, depending on the application, the quadratic model might be preferred because of its simplicity.

x	$SkD_\infty = f(x)$	$d_E(x, SkD_\infty)$	$d_c(x, SkD_\infty)$	Figure
SkD_1	$0.9423x + 0.2885$	0.0086	0.012	7.5
SkD_2	$0.8580x + 0.2304$	0.0034	0.008	7.6
SkD_3	$0.8598x + 0.1922$	0.0016	0.005	7.7
SkD_{1000}	$0.9985x + 0.0016$	0.0000	0.000	7.8
SkD_{10000}	$1.0000x + 0.0000$	0.0000	0.000	7.9

Table 7.2: Summary of experiments for the convergence of β-skeleton depth

x	$HD = f(x)$	$d_E(x, HD)$	$d_c(x, HD)$	Figure
SkD_1	$2.9275(x + 0.0152)^2$	0.0127	0.0306	7.10
SkD_2	$2.5522(x - 0.0608)^2$	0.0141	0.0331	7.11
SkD_3	$2.6569(x - 0.1117)^2$	0.0156	0.0343	7.12
SkD_{1000}	$3.8765(x - 0.2992)^2$	0.0186	0.0373	7.13
SkD_{10000}	$3.8883(x - 0.3004)^2$	0.0186	0.0373	7.14

Table 7.3: Summary of experiments for the halfspace depth approximation via a quadratic model

x	$HD = f(x)$	$d_E(x, HD)$	$d_c(x, HD)$	Figure
SkD_1	$(x + 0.4622)^{4.8174}$	0.0130	0.0306	7.15
SkD_2	$(x - 0.3465)^{4.4263}$	0.0120	0.0331	7.16
SkD_3	$(x + 0.3089)^{4.5734}$	0.0120	0.0343	7.17
SkD_{1000}	$(x + 0.2259)^{5.8737}$	0.0122	0.0373	7.18
SkD_{10000}	$(x + 0.2255)^{5.8850}$	0.0123	0.0373	7.19

Table 7.4: Summary of experiments for the halfspace depth approximation via a power model

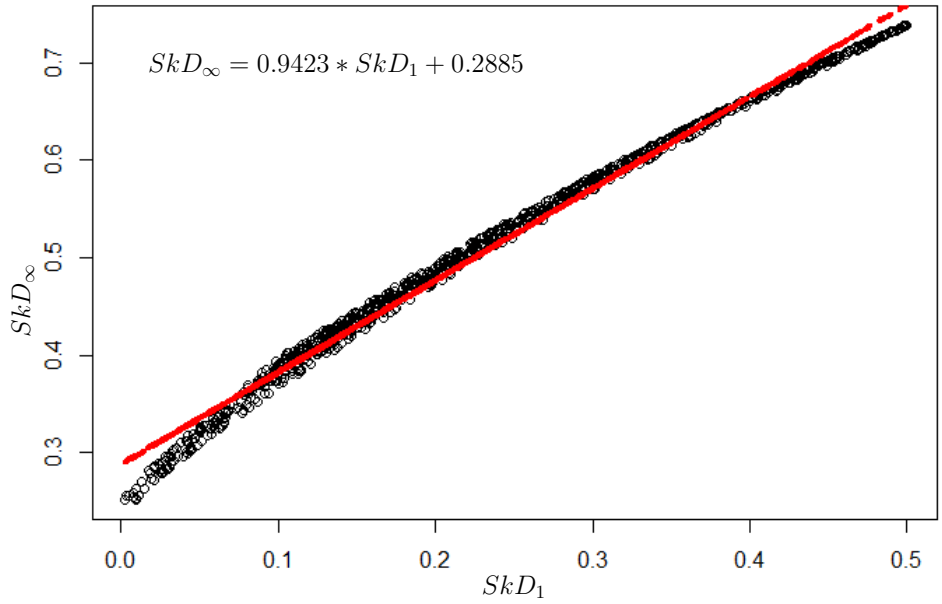

Figure 7.5: Fitting a linear model to SkD_∞ and SkD_1.

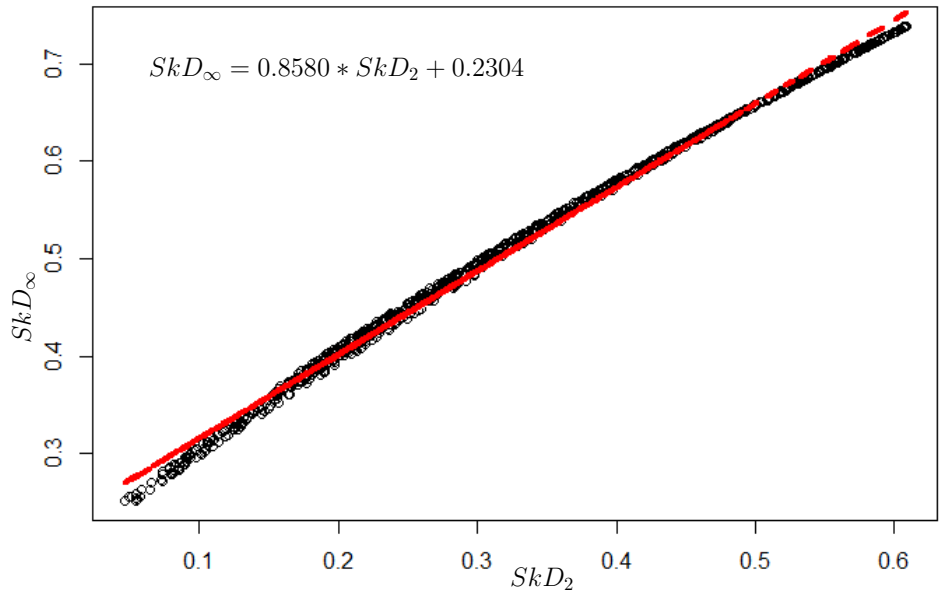

Figure 7.6: Fitting a linear model to SkD_∞ and SkD_2 .

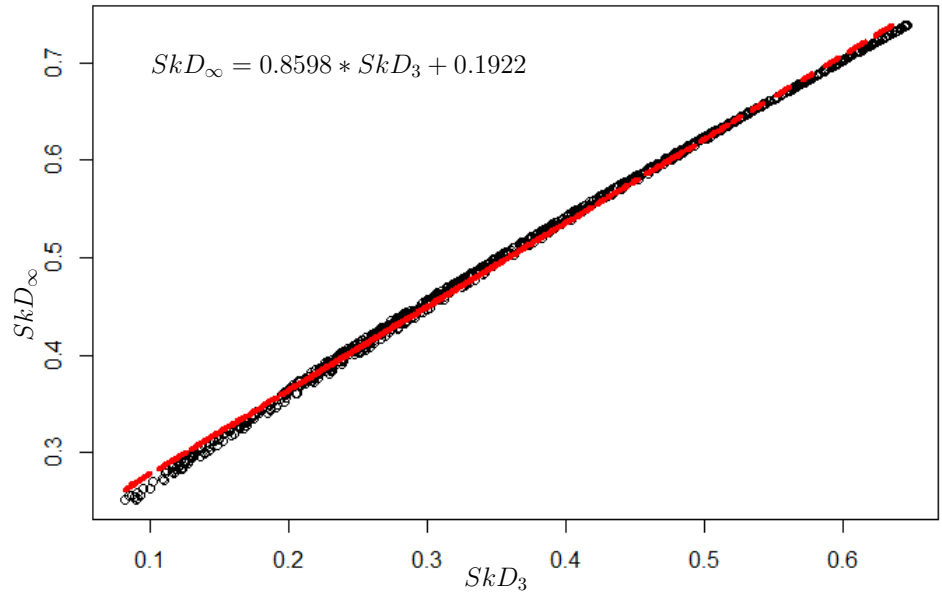

Figure 7.7: Fitting a linear model to SkD_∞ and SkD_3.

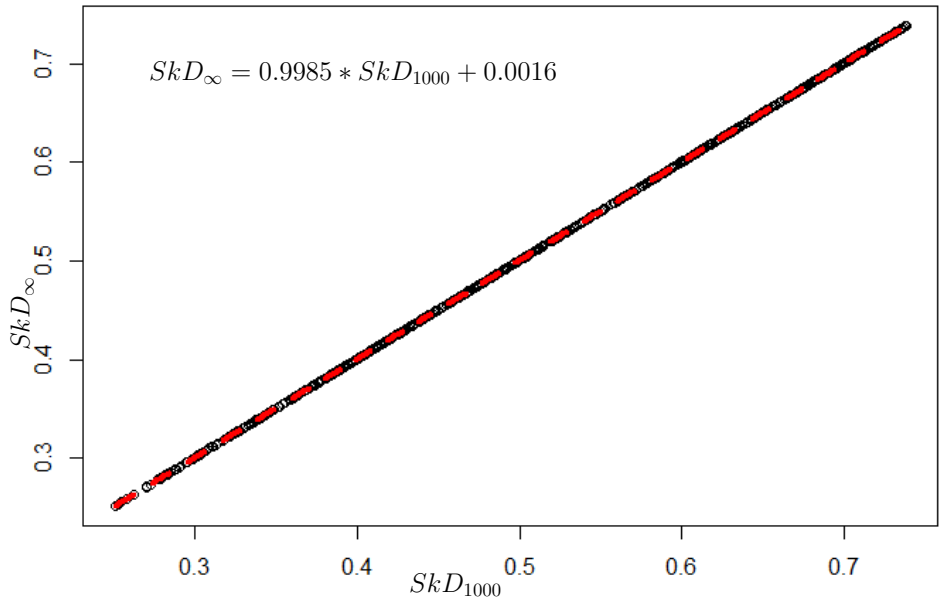

Figure 7.8: Fitting a linear model to SkD_∞ and SkD_{1000}.

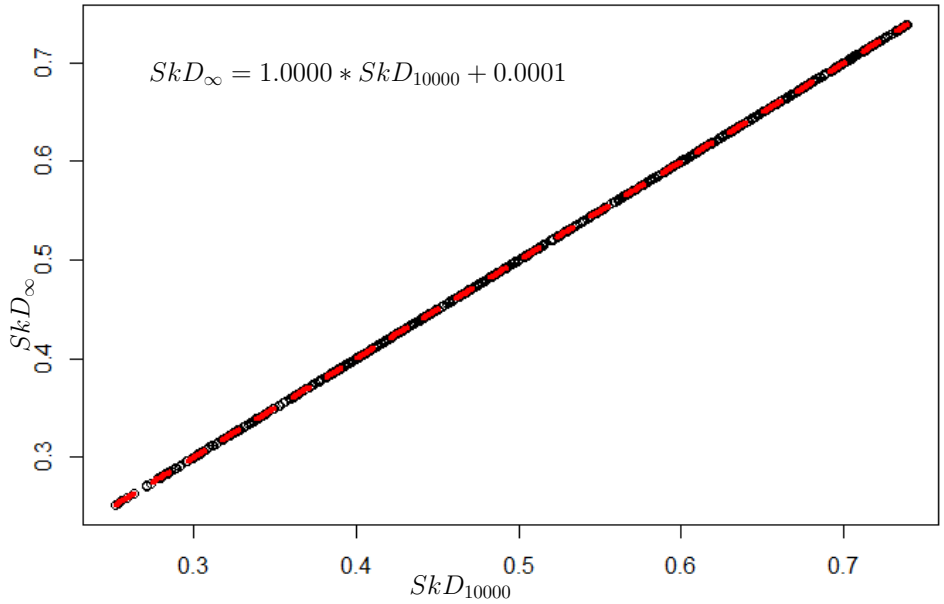

Figure 7.9: Fitting a linear model to SkD_∞ and SkD_{10000}.

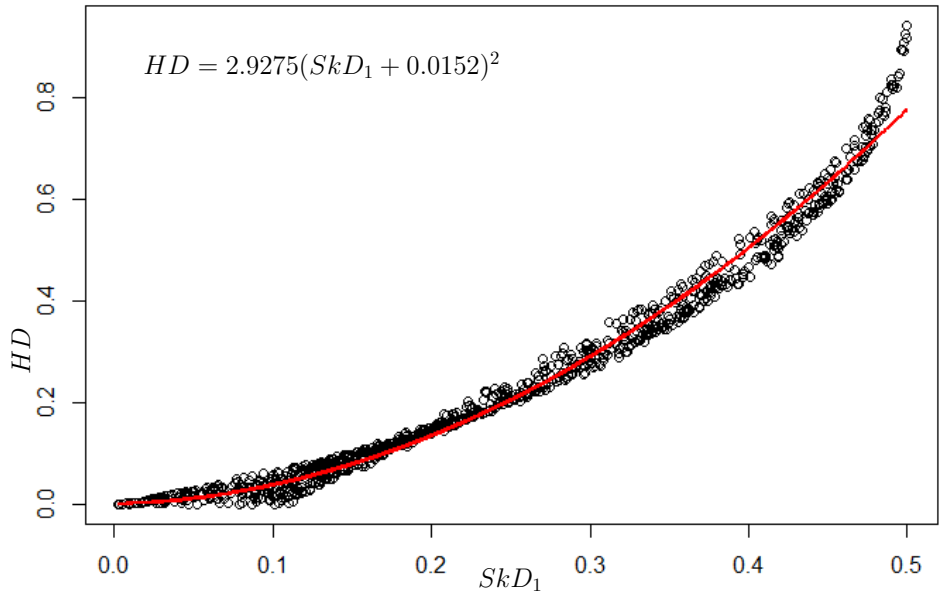

Figure 7.10: Fitting a quadratic model to HD and SkD_1.

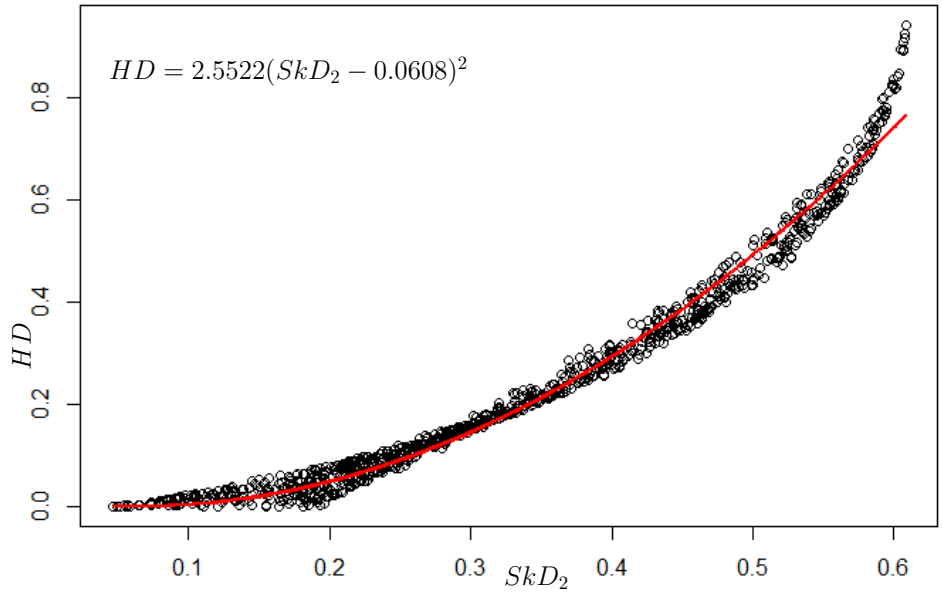

Figure 7.11: Fitting a quadratic model to HD and SkD_2.

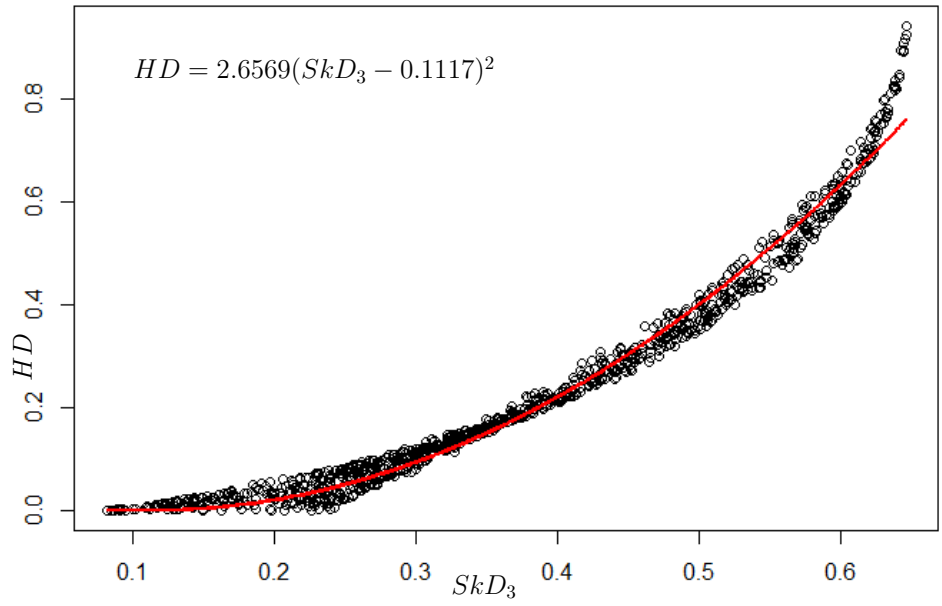

Figure 7.12: Fitting a quadratic model to HD and SkD_3.

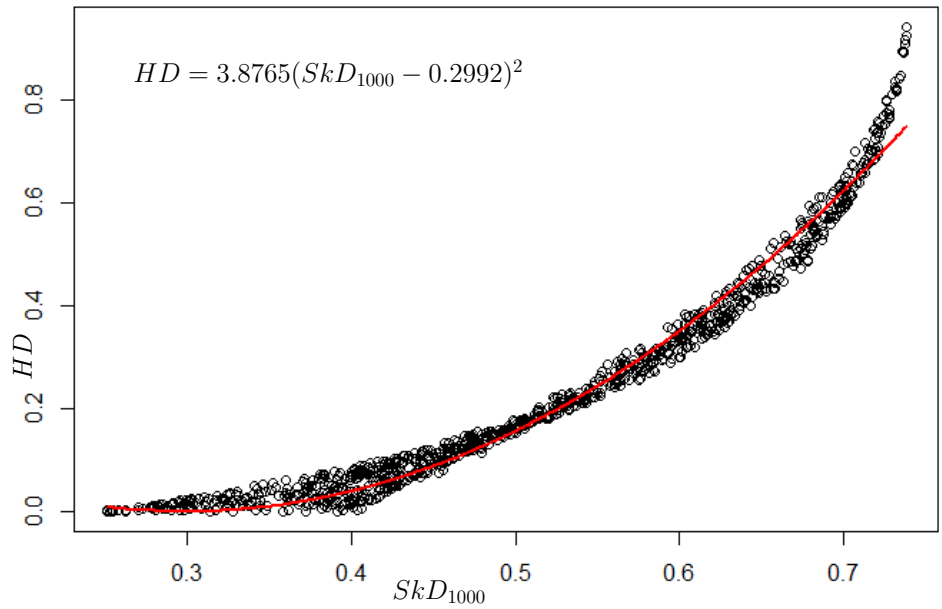

Figure 7.13: Fitting a quadratic model to HD and SkD_{1000}.

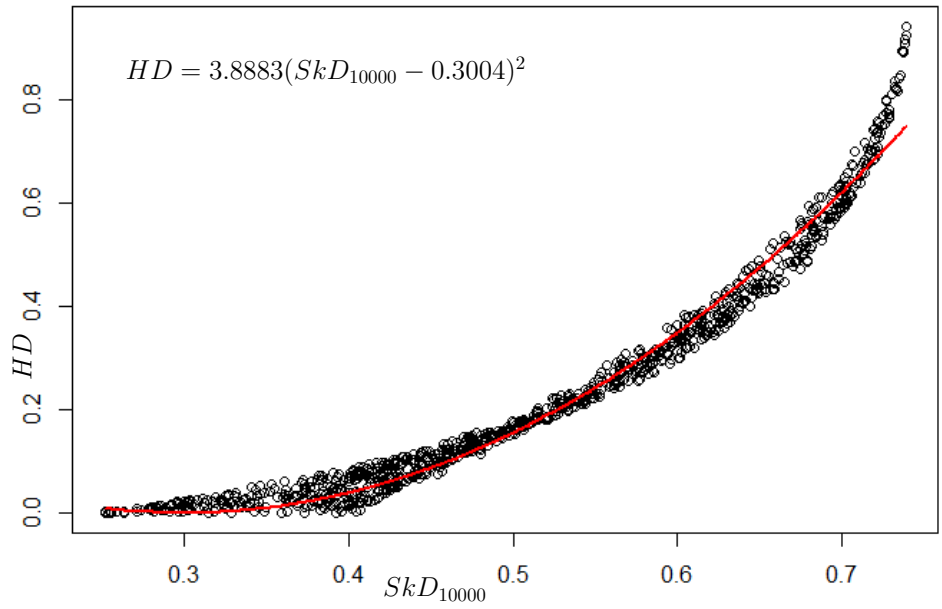

Figure 7.14: Fitting a quadratic model to HD and SkD_{10000}.

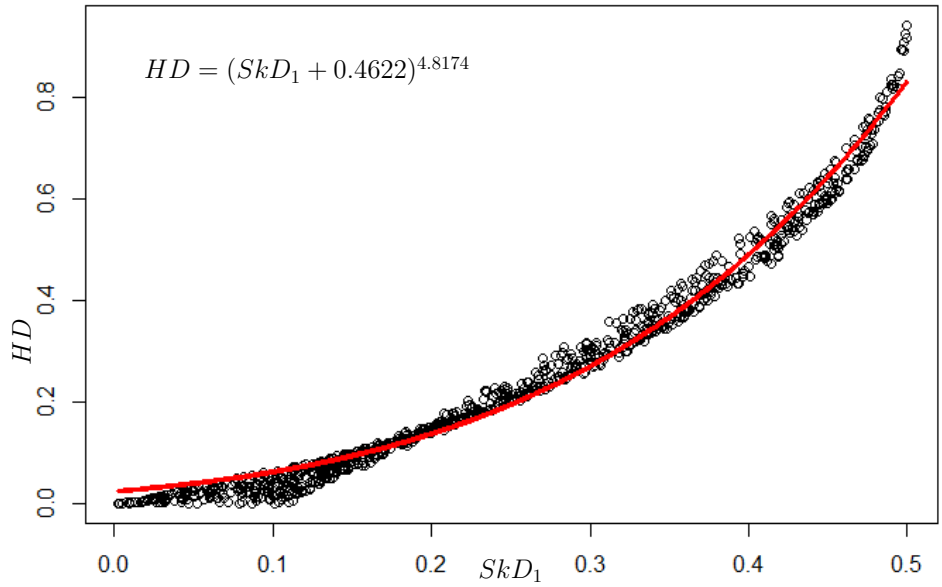

Figure 7.15: Fitting a power model to HD versus SkD_1.

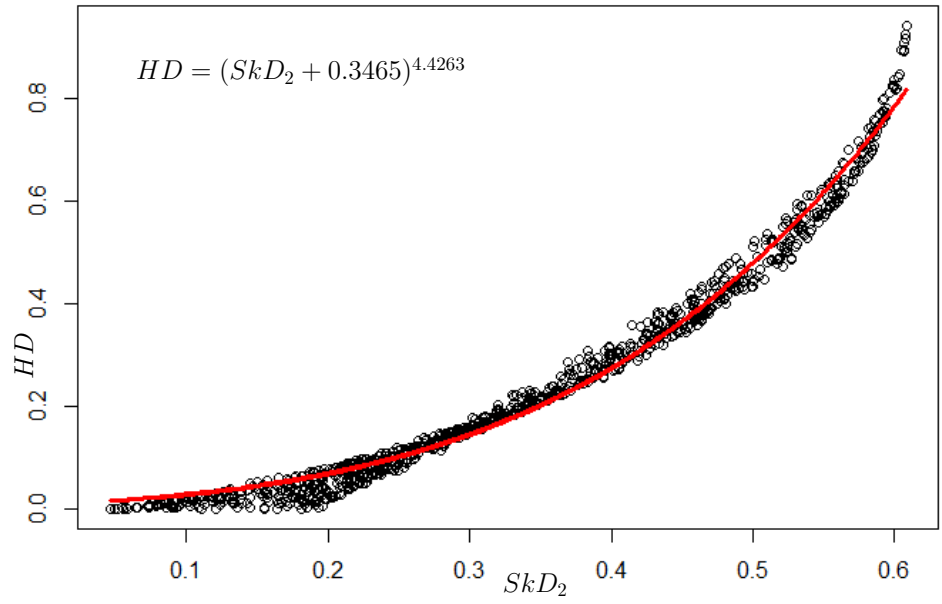

Figure 7.16: Fitting a power model to HD and SkD_2.

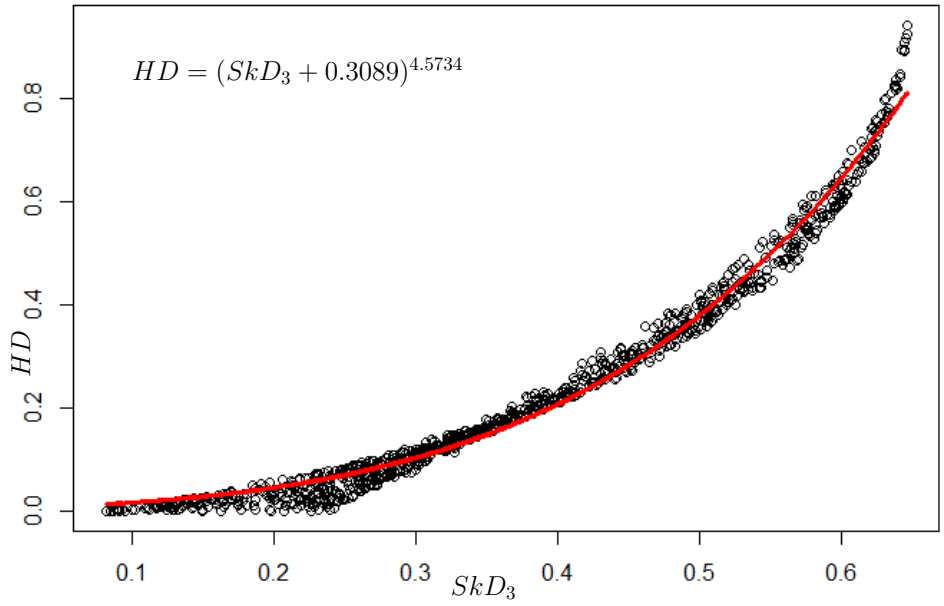

Figure 7.17: Fitting a power model to HD and SkD_3.

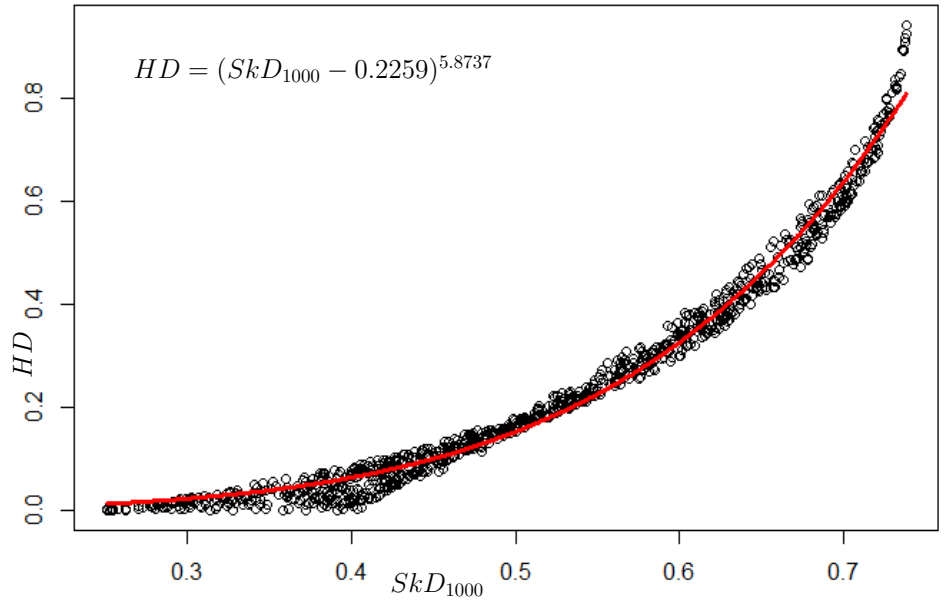

Figure 7.18: Fitting a power model to HD and SkD_{1000}.

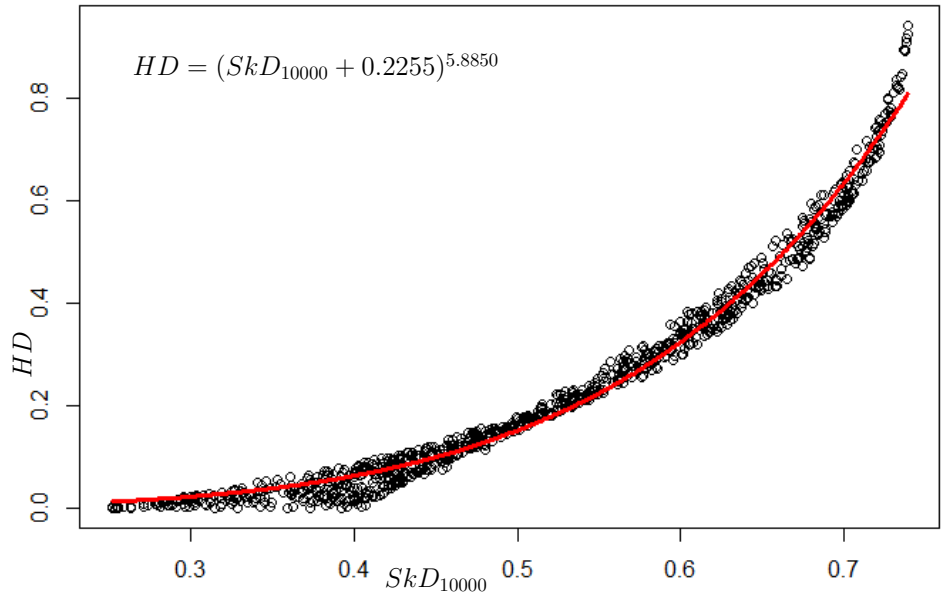

Figure 7.19: Fitting a power model to HD and SkD_{10000}.

Chapter 8

Conclusion

In this thesis we have presented a study of several depth functions such as halfspace depth, simplicial depth, and β-skeleton depth. Emphasis was given to the studying of β-skeleton depth from both geometric and algorithmic viewpoints in $\mathbb{R}^2$. Furthermore, lower bounds for computing the planar β-skeleton depth ($1 \leq \beta \leq \infty$), and approximation of different depth functions are also studied in this thesis. Finally, we provide some experimental results to support our approximation technique, and to illustrate the relationships among the aforementioned depth functions.

8.1 Contributions

Regarding the geometric perspective, in Chapter 4, we proved that the exact bound $(1/4)n^4 - \theta(n^3)$ for the combinatorial complexity of the arrangement

of planar β-influence regions is achievable. The connectivity and convexity of β-skeleton depth regions are also studied in Chapter 4.

For the algorithmic part, in Chapters 5 and 6, we presented an optimal algorithm for computing the planar spherical depth (i.e. β-skeleton depth, $\beta = 1$) of a query point. This algorithm takes $O(n \log n)$ time that matches the corresponding lower bound $\Omega(n \log n)$ proved in Chapter 6. For the other values of $1 < \beta \leq \infty$, employing the results on semialgebraic range counting problems in the literature, we developed an algorithm to compute the planar β-skeleton depth of a query point. This algorithm takes $O(n^{(3/2)+\epsilon})$ time. In developing this algorithm, we reduced the problem of computing the planar β-skeleton depth of a query point to a combination of at most $3n$ range counting problems, where n is the size of the data set (Theorem 5.2.1). In Section 5.3, we presented a simple and optimal algorithm for computing the planar halfspace depth. This algorithm takes $\Theta(n \log n)$ time. There are other optimal algorithms for this problem by Aloupis in [9], and by Chan in [25]. We used our specialized halfspace range counting method to obtain the number of data points in each halfspace whereas in Aloupis's algorithm the idea of sweeping halfline is employed to obtain such number.

The results in Chapter 6 include proving lower bounds for the complexity of computing planar β-skeleton depth, $1 \leq \beta \leq \infty$. In this chapter, using different reductions, we proved that computing the planar β-skeleton depth of

a query point allows us to answer the problem of Element Uniqueness, which takes $\Omega(n \log n)$ time. As such, computing the planar β-skeleton depth also requires $\Omega(n \log n)$ time.

Finally, in Chapter 7, we investigated some relationships among the influence regions of β-skeleton depth and simplicial depth. We employed these relationships to prove that there exists $\beta^* < \infty$ such that for all $\beta \geq \beta^*$, and all q in any finite range $\mathcal{R} \subset \mathbb{R}^2$ that contains S, the values of $SkD_\beta(q; S)$ and $S_\infty(q; S)$ are equal. Also, we proved that the β-skeleton depth has a lower bound in terms of a constant factor of simplicial depth (in particular, $SkD_\beta \geq (2/3)\, SD$). In the remainder of Chapter 7, we proposed a method of approximation, using the idea of fitting functions, to approximate one depth function by another one. To support the theoretical results in Chapter 7, we provided some experimental results. As an example, the experimental results suggest that with a reasonable amount of error, the halfspace depth can be approximated by a quadratic function of the β-skeleton depth ($HD \cong 3.8883(SkD_\beta - 0.3004)^2$ if $\beta \to \infty$).

8.2 Open Problems and Directions for Future Work

Open problem 8.2.1. *In Chapter 6, we proved that planar β-skeleton depth requires $\Omega(n \log n)$ time. However, our best algorithm for computing the pla-*

nar β-skeleton depth ($1 < \beta \leq \infty$) takes $O(n^{(3/2)+\epsilon})$ time. Is it possible to develop an $O(n \log n)$ algorithm for this problem. Note that for the case of $\beta = 1$, we presented such an algorithm.

Open problem 8.2.2. *In studying the relationships among different depth functions, we proved that $SkD_\beta \geq (2/3)\, SD$. Is there some constant c such that for all query points within the convex hull $SkD_\beta \leq c\, SD$.*

Open problem 8.2.3. *In Section 5.2, we used the existing results on semi-algebraic range counting and developed Algorithm 6 with $O(n^{(3/2)+\varepsilon})$ query time, $O(n)$ storage, and $O(n \log n)$ expected preprocessing time. Can the query time be reduced to $O(n^{(4/3)+\varepsilon})$ by spending $O(n^{(4/3)+\varepsilon})$ time and space on the preprocessing. Or, is it possible to answer a semialgebraic range query in $O(n^{(1/3)+\varepsilon})$ time by spending $O(n^{(4/3)+\varepsilon})$ time and space on the preprocessing.*

8.2.1 Future Work

Some of the directions for future work include:

- developing efficient algorithms to compute the β-skeleton median (central point).
- studying the β-skeleton depth in higher dimensions.
- using the β-skeleton depth to find outliers in a data set. This direction can be pursued to develop an efficient intrusion detection system. Re-

garding this application of β-skeleton depth: Algorithm 5 is used in an intrusion detection method proposed in [79].

Bibliography

[1] Peyman Afshani, *On geometric range searching, approximate counting and depth problems*, (2008).

[2] Peyman Afshani and Timothy M Chan, *On approximate range counting and depth*, Discrete & Computational Geometry **42** (2009), no. 1, 3–21.

[3] Pankaj K Agarwal, Jeff Erickson, et al., *Geometric range searching and its relatives*, Contemporary Mathematics **223** (1999), 1–56.

[4] Pankaj K Agarwal, Jiri Matousek, and Micha Sharir, *On range searching with semialgebraic sets II*, SIAM Journal on Computing **42** (2013), no. 6, 2039–2062.

[5] Pankaj K Agarwal and Micha Sharir, *Arrangements and their applications*, Handbook of computational geometry, Elsevier, 2000, pp. 49–119.

[6] Hopcroft Aho, *Ullman*, Data Structures and Algorithms (1974), 204–208.

[7] Htrotugu Akaike, *Maximum likelihood identification of gaussian autoregressive moving average models*, Biometrika **60** (1973), no. 2, 255–265.

[8] Reginald BJT Allenby and Alan Slomson, *How to count: An introduction to combinatorics*, CRC Press, 2011.

[9] Greg Aloupis, *On computing geometric estimators of location, 2001*, Master's thesis, M. Sc. , McGill University.

[10] ———, *On computing geometric estimators of location*, Ph.D. thesis, McGill University, 2001.

[11] ———, *Geometric measures of data depth*, DIMACS series in discrete mathematics and theoretical computer science **72** (2006), 147.

[12] Greg Aloupis, Carmen Cortés, Francisco Gómez, Michael Soss, and Godfried Toussaint, *Lower bounds for computing statistical depth*, Computational Statistics & Data Analysis **40** (2002), no. 2, 223–229.

[13] Greg Aloupis, Stefan Langerman, Michael Soss, and Godfried Toussaint, *Algorithms for bivariate medians and a fermat–torricelli problem for lines*, Computational Geometry **26** (2003), no. 1, 69–79.

[14] Boris Aronov and Sariel Har-Peled, *On approximating the depth and related problems*, SIAM Journal on Computing **38** (2008), no. 3, 899–921.

[15] Boris Aronov and Micha Sharir, *Approximate halfspace range counting*, SIAM Journal on Computing **39** (2010), no. 7, 2704–2725.

[16] Vic Barnett, *The ordering of multivariate data*, Journal of the Royal Statistical Society. Series A (General) (1976), 318–355.

[17] Gilbert W Bassett Jr, *Equivariant, monotonic, 50% breakdown estimators*, The American Statistician **45** (1991), no. 2, 135–137.

[18] Ayanendranath Basu, Ian R Harris, Nils L Hjort, and MC Jones, *Robust and efficient estimation by minimising a density power divergence*, Biometrika **85** (1998), no. 3, 549–559.

[19] Michael Ben-Or, *Lower bounds for algebraic computation trees*, Proceedings of the Fifteenth annual ACM Symposium on Theory of computing, ACM, 1983, pp. 80–86.

[20] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars, *Computational geometry: algorithms and applications*, Springer-Verlag TELOS, 2008.

[21] David Bremner, Dan Chen, John Iacono, Stefan Langerman, and Pat Morin, *Output-sensitive algorithms forTukey depth and related problems*, Statistics and Computing **18** (2008), no. 3, 259–266.

[22] David Bremner, Komei Fukuda, and Vera Rosta, *Primal-dual algorithms for data depth*, Diamacs Series in discrete mathematics and theoretical computer science **72** (2006), 171.

[23] Michael A Burr, Eynat Rafalin, and Diane L Souvaine, *Dynamic maintenance of half-space depth for points and contours*, arXiv preprint arXiv:1109.1517 (2011).

[24] Constantin Carathéodory, *Über den variabilitätsbereich der koeffizienten von potenzreihen, die gegebene werte nicht annehmen*, Mathematische Annalen **64** (1907), no. 1, 95–115.

[25] Timothy M Chan, *An optimal randomized algorithm for maximum Tukey depth*, Proceedings of the Fifteenth annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2004, pp. 430–436.

[26] Dan Chen, *Algorithms for data depth*, Carleton University, 2013.

[27] Dan Chen, Pat Morin, and Uli Wagner, *Absolute approximation of Tukey depth: Theory and experiments*, Computational Geometry **46** (2013), no. 5, 566–573.

[28] ZQ Chen, *Bounds for the breakdown point of the simplicial median*, Journal of Multivariate Analysis **55** (1995), no. 1, 1–13.

[29] Andrew Y Cheng and Ming Ouyang, *On algorithms for simplicial depth.*, CCCG, 2001, pp. 53–56.

[30] Andreas Christmann, *Regression depth and support vector machine*, DIMACS series in discrete mathematics and theoretical computer science **72** (2006), 71.

[31] Juan Antonio Cuesta-Albertos and Alicia Nieto-Reyes, *The random Tukey depth*, Computational Statistics & Data Analysis **52** (2008), no. 11, 4979–4988.

[32] David P Dobkin and Richard J Lipton, *On the complexity of computations under varying sets of primitives*, Journal of Computer and System Sciences **18** (1979), no. 1, 86–91.

[33] D Donoho, *Breakdown properties of multivariate location estimators." Ph. D. qualifying paper, Harvard University*, (1982).

[34] David L Donoho and Miriam Gasko, *Breakdown properties of location estimates based on halfspace depth and projected outlyingness*, The Annals of Statistics (1992), 1803–1827.

[35] David L Donoho and Peter J Huber, *The notion of breakdown point*, A festschrift for Erich L. Lehmann **157184** (1983).

[36] Stephane Durocher and David Kirkpatrick, *The projection median of a set of points*, Computational Geometry **42** (2009), no. 5, 364–375.

[37] Rainer Dyckerhoff, Karl Mosler, and Gleb Koshevoy, *Zonoid data depth: Theory and computation*, COMPSTAT, Springer, 1996, pp. 235–240.

[38] Ryan T Elmore, Thomas P Hettmansperger, and Fengjuan Xuan, *Spherical data depth and a multivariate median*, DIMACS Series in Discrete Mathematics and Theoretical Computer Science **72** (2006), 87.

[39] Marco Fattore, Rosanna Grassi, and Alberto Arcagni, *Measuring structural dissimilarity between finite partial orders*, Multi-indicator Systems and Modelling in Partial Order, Springer, 2014, pp. 69–84.

[40] Stefan Felsner, *Geometric graphs and arrangements: some chapters from combinatorial geometry*, Springer Science & Business Media, 2012.

[41] K Ruben Gabriel and Robert R Sokal, *A new statistical approach to geographic variation analysis*, Systematic Zoology **18** (1969), no. 3, 259–278.

[42] Neil A Gershenfeld and Neil Gershenfeld, *The nature of mathematical modeling*, Cambridge university press, 1999.

[43] Jacob E Goodman, Richard D Pollack, and William L Steiger, *Discrete and computational geometry: papers from the dimacs special year*, vol. 6, American Mathematical Soc., 1991.

[44] Sariel Har-Peled, *Geometric approximation algorithms*, no. 173, American Mathematical Soc., 2011.

[45] Sariel Har-Peled and Micha Sharir, *Relative (p, ε)-approximations in geometry*, Discrete & Computational Geometry **45** (2011), no. 3, 462–496.

[46] Michiel Hazewinkel, *Encyclopaedia of mathematics: C an updated and annotated translation of the soviet mathematical encyclopaedia*, vol. 2, Springer Science & Business Media, 2013.

[47] Xuming He and Gang Wang, *Convergence of depth contours for multivariate datasets*, The Annals of Statistics (1997), 495–504.

[48] Harold Hotelling, *Stability in competition*, The Collected Economics Articles of Harold Hotelling, Springer, 1990, pp. 50–63.

[49] Oleg A Ivanov, *Making mathematics come to life: A guide for teachers and students*, American Mathematical Soc., 2009.

[50] David S. Johnson and Franco P Preparata, *The densest hemisphere problem*, Theoretical Computer Science **6** (1978), no. 1, 93–107.

[51] Stefan Langerman and William Steiger, *The complexity of hyperplane depth in the plane*, Symposium on Discrete Algorithms, ACM and SIAM, 2000.

[52] ———, *Optimization in arrangements*, Annual Symposium on Theoretical Aspects of Computer Science, Springer, 2003, pp. 50–61.

[53] Stefan Langerman False Swarzberg, *Algorithms and data structures in computational geometry*, (2001).

[54] Regina Y Liu, *On a notion of data depth based on random simplices*, The Annals of Statistics (1990), 405–414.

[55] ———, *Control charts for multivariate processes*, Journal of the American Statistical Association **90** (1995), no. 432, 1380–1387.

[56] Regina Y Liu, Robert Joseph Serfling, and Diane L Souvaine, *Data depth: robust multivariate analysis, computational geometry, and applications*, vol. 72, American Mathematical Soc., 2006.

[57] Regina Y Liu and Kesar Singh, *A quality index based on data depth and multivariate rank tests*, Journal of the American Statistical Association **88** (1993), no. 421, 252–260.

[58] Zhenyu Liu and Reza Modarres, *Lens data depth and median*, Journal of Nonparametric Statistics **23** (2011), no. 4, 1063–1074.

[59] Hendrik P Lopuhaa, *Highly efficient estimators of multivariate location with high breakdown point*, The Annals of Statistics (1992), 398–413.

[60] Hendrik P Lopuhaa and Peter J Rousseeuw, *Breakdown points of affine equivariant estimators of multivariate location and covariance matrices*, The Annals of Statistics **19** (1991), no. 1, 229–248.

[61] Prasanta Chandra Mahalanobis, *On the generalized distance in statistics*, National Institute of Science of India, 1936.

[62] Jiri Matousek, *Computing the center of planar point sets*, Discrete and Computational Geometry **6** (1991), 221.

[63] Kim Miller, Suneeta Ramaswami, Peter Rousseeuw, Toni Sellarès, Diane Souvaine, Ileana Streinu, and Anja Struyf, *Fast implementation of*

depth contours using topological sweep, Proceedings of the Twelfth annual ACM-SIAM Symposium on Discrete Algorithms, Society for Industrial and Applied Mathematics, 2001, pp. 690–699.

[64] Reza Modarres and Mengta Yang, *β-skeleton depth functions and medians*, Department of Statistics, The George Washington University, Washington (2015), no. 131.

[65] Karl Mosler, *Multivariate dispersion, central regions, and depth: The lift zonoid approach*, vol. 165, Springer Science & Business Media, 2012.

[66] ———, *Depth statistics*, Robustness and complex data structures, Springer, 2013, pp. 17–34.

[67] Jay I Myung, Daniel R Cavagnaro, and Mark A Pitt, *Model evaluation and selection*, New handbook of mathematical psychology **1** (2016).

[68] Hannu Oja, *Descriptive statistics for multivariate distributions*, Statistics & Probability Letters **1** (1983), no. 6, 327–332.

[69] Franco P Preparata and Michael I Shamos, *Computational geometry: an introduction*, Springer Science & Business Media, 2012.

[70] Michael O Rabin, *Proving simultaneous positivity of linear forms*, Journal of Computer and System Sciences **6** (1972), no. 6, 639–650.

[71] Edward M Reingold, *On the optimality of some set algorithms*, Journal of the ACM (JACM) **19** (1972), no. 4, 649–659.

[72] Peter J Rousseeuw and Mia Hubert, *Regression depth*, Journal of the American Statistical Association **94** (1999), no. 446, 388–402.

[73] Peter J Rousseeuw and Ida Ruts, *Algorithm as 307: Bivariate location depth*, Journal of the Royal Statistical Society. Series C (Applied Statistics) **45** (1996), no. 4, 516–526.

[74] ———, *Constructing the bivariate Tukey median*, Statistica Sinica (1998), 827–839.

[75] Peter J Rousseeuw and Anja Struyf, *Computing location depth and regression depth in higher dimensions*, Statistics and Computing **8** (1998), no. 3, 193–203.

[76] Gideon Schwarz et al., *Estimating the dimension of a model*, The Annals of Statistics **6** (1978), no. 2, 461–464.

[77] Douglas S Shafer and Z Zhang, *Beginning statistics*, Phylis-Barnidge publisher **304** (2012).

[78] Michael Ian Shamos, *Geometry and statistics: Problems at the interface*, In Algorithms and Complexity, Citeseer, 1977.

[79] Iman Sharafaldin and Ali A Ghorbani, *Eagleeye: A novel visual anomaly detection method*, 2018 16th Annual Conference on Privacy, Security and Trust (PST), IEEE, 2018, pp. 1–6.

[80] Hayim Shaul, *Range searching: Emptiness, reporting, and approximate counting*, University of Tel-Aviv, 2011.

[81] Christopher G Small, *A survey of multidimensional medians*, International Statistical Review/Revue Internationale de Statistique (1990), 263–277.

[82] J Michael Steele and Andrew C Yao, *Lower bounds for algebraic decision trees*, Journal of Algorithms **3** (1982), no. 1, 1–8.

[83] Anja J Struyf and Peter J Rousseeuw, *Halfspace depth and regression depth characterize the empirical distribution*, Journal of Multivariate Analysis **69** (1999), no. 1, 135–153.

[84] Kenneth J Supowit, *The relative neighborhood graph, with an application to minimum spanning trees*, Journal of the ACM (JACM) **30** (1983), no. 3, 428–448.

[85] James Tanton, *Encyclopedia of mathematics.*, Facts On File, inc., 2005.

[86] John W Tukey, *Mathematics and the picturing of data*, Proceedings of the international congress of mathematicians, vol. 2, 1975, pp. 523–531.

[87] R Wilcox, *Approximating Tukey's depth*, Communications in Statistics-Simulation and Computation **32** (2003), no. 4, 977–985.

[88] Mengta Yang, *Depth functions, multidimensional medians and tests of uniformity on proximity graphs*, Ph.D. thesis, The George Washington University, 2014.

[89] Mengta Yang and Reza Modarres, *β-skeleton depth functions and medians*, Communications in Statistics-Theory and Methods (2017), 1–17.

[90] Yijun Zuo and Robert Serfling, *General notions of statistical depth function*, Annals of Statistics (2000), 461–482.

Vita

Candidate's full name: Rasoul Shahsavarifar
University attended (with dates and degrees obtained):
University of new Brunswick, Fredericton, Canada, 2014-2019, PhD
University of Tabriz, Tabriz, Iran, 2006-2008, MSc
Razi University, Kermanshah, Iran, 2001-2006, BSc

Publications:
Hassan Mahdikhani, Rasoul Shahsavarifar, Rongxing Lu, David Bremner, Achieve Privacy-Preserving Simplicial Depth Query over Outsourced Cloud Platform, manuscript submitted to Journal of Information Security and Applications, March 2019.
Rasoul Shahsavarifar and David Bremner, " Approximate Data depth Revisited", CCCG2018, University of Manitoba, Winnipeg, MB, Canada, 7-10 Aug 2018.
Rasoul Shahsavarifar and David Bremner, " Approximate Data Depth Revisited", arXiv preprint arXiv:1805.07373, 2018.
Rasoul Shahsavarifar and David Bremner, " Computing the Planar β-skeleton Depth", arXiv preprint arXiv:1803.05970, 2018.
Rasoul Shahsavarifar and David Bremner, " An Optimal Algorithm for Computing the Spherical Depth of Points in the Plane", arXiv preprint arXiv:1702.07399, 2017.

Conference Presentations:
Rasoul Shahsavarifar and David Bremner, " Approximate Data depth Revisited", CCCG2018, University of Manitoba, Winnipeg, MB, Canada, 7-10 Aug 2018.

Rasoul Shahsavarifar and David Bremner, " Approximation of Data depth", CMS2018, University of New Brunswick, Fredericton, NB, Canada, 1-4 June 2018.

Rasoul Shahsavarifar and David Bremner, " On the Planar Spherical Depth and Lens Depth", CCCG2017, Carleton university, Ottawa, ON, Canada, 2017.

Rasoul Shahsavarifar and David Bremner, " Computing the spherical depth of points in the plane", FWCG2016, University of New York, New York,USA, 2016.

Rasoul Shahsavarifar and David Bremner, " Geometric and Computational Aspects of Data Depth", Annual Graduate Research Conference, University of New Brunswick, Fredericton, NB, Canada, April 23, 2015.