MAXIMAL AND CONVEX LAYERS OF RANDOM POINT SETS

by

Cuong Phu Nguyen

Submitted in partial fulfillment of the requirements
for the degree of Master of Computer Science

at

Dalhousie University
Halifax, Nova Scotia
October 2017

This thesis is dedicated to my wife, who has been patiently waiting until it is finished.

Table of Contents

List of Figures

Abstract

This thesis studies two well-known geometric structures in computational geometry: maximal points and convex hull. Extending the concepts to multiple maximal and convex layers is natural. We study the maximal and convex layers of a point set in d dimensions drawn from a uniform or component-independent (CI) distribution. A distribution is component-independent if each coordinate of a point is chosen independently from continuous distribution. Precisely, we want to compute and to bound the expected size of the first k layers. For the first set of results, we show that, for $d \in \{2, 3\}$, the first $n^{1/d-\epsilon}$ maximal layers can be computed using $dn + o(n)$ scalar comparisons with high probability. For $d \geq 4$, the first $n^{1/2d-\epsilon}$ maximal layers can be computed within this bound with high probability. The first $n^{1/d-\epsilon}$ convex layers in two dimensions, the first $n^{1/2d-\epsilon}$ convex layers in 3D, and the first $n^{1/(d^2+2)}$ convex layers in $d \geq 4$ dimensions can be computed using $2dn + o(n)$ scalar comparisons with high probability. Since the expected number of maximal layers in 2D is $2\sqrt{n}$, our result for 2D maximal layers shows that it takes $dn + o(n)$ scalar comparisons to compute a $1/n^{\epsilon}$-fraction of all layers in the average case. For the second set of results, we show that the kth maximal and convex layer of a point set drawn from a continuous CI distribution in d dimensions has expected size $O(k^d \log^{d-1}(n/k^d))$.

List of Abbreviations and Symbols Used

GIS	Geographic Information Systems
IC	Integrated Circuit
LMIS	Longest Monotonically Increasing Subsequence
CI	Component Independent
$\nearrow$	Dominate
$SL^{(k)}(S)$	The set of the first k maximal layers of the set S
$p \circ q$	The Hadamard Product
$CH^{(k)}(S)$	The set of the first k convex layers of the set S
$OH^{(k)}(S)$	The set of the first k orthant layers of the set S

Acknowledgements

I would like to express my enormous gratitude to my super cool and smart supervisors, Dr. Meng He and Dr. Norbert Zeh, for their mentorship and support throughout many years. I do improve the critical thinking skill a lot by working with them.

In addition, I want to give my gratitude to Professor Jeannette Janssen and Dr. Alex Brodsky who have kindly given me their insightful suggestions as the readers. I also want to sincerely thank the Faculty of Computer Science, Dalhousie University for being a wonderful place and having all kind, generous staff and students. I will miss this place so much. I am and will always be in great debt to my supervisors, the university, and many other kind people in Canada.

Last but not least, I am very grateful to the anonymous reviewer, without whom we would not complete the second set result.

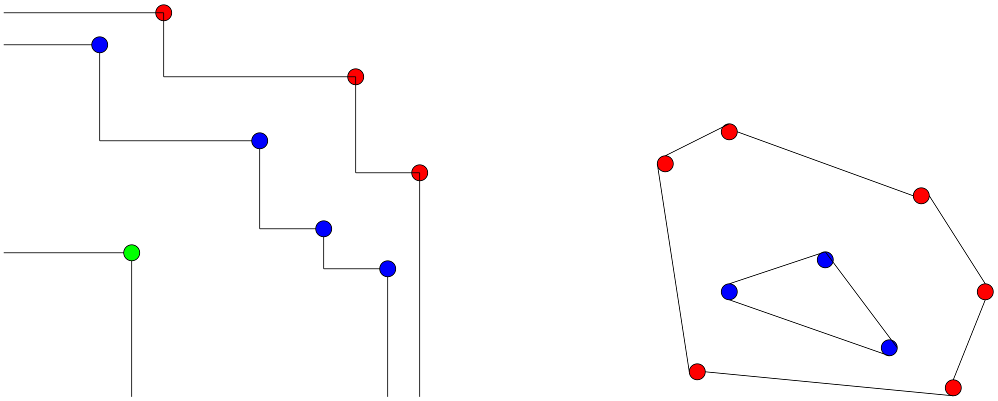
Chapter 1

Introduction

Computational geometry is a branch of computer science studying algorithms dealing with geometry structures. Computational geometry started in the late 1970s [4] and has grown into a large, thriving research field to which many high-quality journals and conferences are solely devoted. Computational complexity is the center of computational geometry where the practical application could run on large datasets ranging from tens to hundreds of millions of points [8]. The application domains include, but are not limited to, computer graphics, geometrical searching and route planning in geographic information systems (GIS), motion planning and collision detection in robotics, and design and verification of integrated circuits (IC).

Among others, maximal and convex layers are two fundamental geometric structures in computational geometry. For two distinct points p and q, point p dominates q if p's coordinate is at least q's coordinate in every dimension. For a point set S in d-dimensional space, $p \in S$ is called a *maximal* point if there is no point in S dominating p. The *skyline* (*first maximal layer*) of S is the set of all *maximal* points in S. Refer to Figure 1.1 (a), the skyline contains all the red points. For an alternative definition, the *orthant* at a point p is the region containing all points dominated by p. The *skyline* is the set of points both in S and on the boundary of the union of the orthants of all $p \in S$ [1]. Finding the *skyline* of a point set has a broad range of applications in statistic, economics, and operations research [38]. It is also used to compute the running time of a dynamic algorithm [2] and to solve the floating currency problem in economics [38]. In the database community, the standard SQL *skyline* operator is computing maximal points under the hood [6].

A point $p \in S$ belongs to the *convex hull* (*first convex layer*) of S if there exists a $(d-1)$-dimensional hyperplane through p that has all points of S on the same side. Refer to Figure 1.1 (b), the convex hull contains all the red points. *Convex hull* is extensively used in computer visualization, GIS and pattern recognition. For $k > 1$, the *kth maximal or convex layer* is the skyline or convex hull of the subset of S obtained by removing the first



(a) Maximal layers

(b) Convex layers

Figure 1.1: Maximal layers and Convex layers in two dimensional space.

$k-1$ maximal or convex layers, respectively. Finding multiple *maximal or convex layers* has many applications in data mining [6], pattern recognition and statistics [12, 35]. The optimal solution to the half-plane range search problem, a standard retrieval problem, depends on convex layers [15]. Computing convex layers is also an interesting problem on its own because intuitively it is *geometrically equivalent* to sorting problem [12]. Although it is commonly known that sorting can be reduced to convex hull, computing only the convex hull of an arbitrary point set is often easier than sorting. When the size of the convex hull is a small fraction of the size of the point set, a much faster algorithm can be designed to compute the convex hull [3]. By computing all convex layers, we *lift* the complexity of these problems to catch up with sorting.

1.1 Motivation

As we shall discuss in more detail in Chapter 2, maximal and convex layers are geometric structures that have been studied extensively. So far, only the expected sizes, for a particular set of distributions, of the first maximal and convex layers are known [2]. Although the technique in [2] works well for point sets in general d-dimensional space, it is unlikely that the same technique can be extended to bound the size of any inner layers, because the analysis heavily depends on the component-independent property of the input.

Besides the complexity aspect, computing multiple maximal and convex layers is interesting on its own. In Chapter 2, we will see that there have been many optimal results in the worst-case analysis for small dimensional space ($d = 2, 3$). For $d = 2$, researchers even achieved output-sensitive results [35], in which the running time depends on the size of the output. It is not hard to come up with an adversary argument showing that any algorithm correctly computing even only the first maximal or convex layer in $d \geq 2$ dimensions requires at least $\Omega(n \log n)$ scalar comparisons in the worst case. Scalar comparisons are the comparisons between two coordinates or between one coordinate and one fixed value. On the other hand, for component-independent (CI) point distributions, one can compute the skyline or convex hull in expected linear time. A point distribution is component-independent if each coordinate of a point is chosen independently from continuous distribution. The restriction to CI distribution is not "too narrow" because CI distribution is the only distribution studied in many well-known papers, for example, [19], [2], [25]. In fact, Dalal [19] only deals with uniform distribution, a particular type of CI distribution.

Comparison-based algorithms are the algorithms whose decisions are made by the results of scalar comparisons. A natural goal is to minimize the exact constant factors in the leading term of the number of comparisons, as done for sorting [22, 30], median finding [20, 21], heap building [26, 32], and orthogonal type problems [11]. To that end, Bentley et al. [3] presented an algorithm to find the skyline of a point set in d dimensions using only $dn + o(n)$ expected scalar comparisons. For convex hull, Bentley et al.'s algorithm [3] used $2dn + o(n)$ expected scalar comparisons for $d \in \{2, 3\}$. For $d \geq 4$, the algorithm found a superset of input points, staying on the boundary of the convex hull, of expected size $O(\log^{d-1} n)$ using $2dn + o(n)$ scalar comparisons in expectation. In [3], dn is proven to be the lower bound on the number of scalar comparisons required for computing either the skyline or the convex hull.

There have been several follow-up results from [3], such as the improvement of lower order terms in [18]. The work of Golin [25] is also interesting because it shows that the alternative algorithm to compute the skyline, proposed in [3], in fact uses $n + o(n)$ expected scalar comparisons for inputs from any two-dimensional CI distribution, though its running time was not analyzed in [3]. Golin [25] also lifted the result in expectation to high probability. Up to now, only expected bounds on the number of scalar comparisons were known even for computing only the first convex or maximal layer, with the exception of [25]. However, it is unclear how to extend the work from [25] to higher dimensions.

1.2 Our contribution

This thesis extends Bentley et al.'s results to multiple layers in both expectation and with high probability. Our results are the first to match the optimal constant factor in the leading term of the expected number of scalar comparisons on random point sets when extracting more than one maximal layers. In the case of the convex hull, the constant factor is up to two times the optimal one. Remarkably, in high dimensional space where $d \geq 4$, we are the first to compute multiple convex layers on random point sets in linear time. In fact, no $O(n \operatorname{polylog}(n))$ algorithm is known so far. For a set S of n planar points sorted by the x-coordinates, the number of maximal layers of S is the length of a longest monotonically increasing subsequence (LMIS) of the sequence of y-coordinates in S. If the distribution from which S is drawn is CI, this sequence of y-coordinates is a uniform random permutation of the y-coordinates. From [23], the expected length of an LMIS of this sequence approximates to $2\sqrt{n}$ when n is large enough. Thus at least in two dimensions, since our algorithm finds $n^{1/2-\epsilon}$ maximal layers, we can extract a $1/n^{\epsilon}$-fraction of all maximal layers while using a *near* optimal expected number of scalar comparisons.

Although it is not difficult to extend Bentley et al.'s algorithms to compute inner layers, we find that it is more challenging to analyze the running time. The fundamental idea of our analysis involves a process of conceptually subdividing an object into smaller objects that are *similar* in term of topology. The existence of an input point inside each similar object will guarantee the *topology* of one of the first layers. The idea may be of general interest, as it may be useful to analyze similar problems involving multiple layers. For example, section 4.3 uses the idea to analyze multiple orthant layers. Indeed, we apply the above idea to establish upper bounds on the expected size of maximal and convex layers of random point sets.

Our first main results transform a worst-case algorithm computing layers in d dimensions in $O(k^c n^{1+\epsilon})$ scalar comparisons into an algorithm computing the first $n^{\frac{1}{c+1}-\epsilon}$ maximal or convex layers, but using only $dn + o(n)$ or $2dn + o(n)$ scalar comparisons, respectively, in expectation and with high probability, where c and ϵ are constants with $c \geq 0$ and $0 < \epsilon < \frac{1}{(c+1)d}$. With this transformation, we obtained the following results:

1. We provide an algorithm to compute the first $k = n^{\frac{1}{d}-\epsilon}$ maximal layers using only $dn+o(n)$ scalar comparisons in expectation and with high probability, where $d \in \{2, 3\}$.

2. We provide an algorithm to compute the first $k = n^{\frac{1}{2d}-\epsilon}$ maximal layers using only $dn + o(n)$ scalar comparisons in expectation and with high probability, where $d \geq 4$.

3. We provide an algorithm to compute the first $k = n^{\frac{1}{d}-\epsilon}$ convex layers using only $2dn + o(n)$ scalar comparisons in expectation and with high probability, where $d = 2$.

4. We provide an algorithm to compute the first $k = n^{\frac{1}{2d}-\epsilon}$ convex layers using only $2dn + o(n)$ scalar comparisons in expectation and with high probability, where $d = 3$.

5. We provide an algorithm to compute the first $k = n^{1/(d^2+2)}$ convex layers using only $2dn + o(n)$ scalar comparisons with high probability $1 - O(\frac{1}{n^{1/d-\epsilon}})$, where $d \geq 4$.

In the first four results, the high probability is equal to $1 - o(n^{-n^{\gamma}})$, where γ is an arbitrary constant in $(0, c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)d})$.

Our second main results bound the size of the k-th maximal or convex layer of a d-dimensional point set. We show that, for any point set S drawn from a continuous CI distribution in d dimensions and a fixed parameter $k \leq n$,

1. The kth maximal layer has expected size $O(k^d \log^{d-1}(n/k^d))$.

2. The kth convex layer has expected size $O(k^d \log^{d-1}(n/k^d))$.

1.3 Outline

Chapter 2 provides a survey on the literature of maximal and convex layers, and some other related computational geometry topics. Chapter 3 proves our first set of results, while Chapter 4 deals with the second set. And finally, Chapter 5 offers concluding remarks.

Chapter 2

Previous work

This chapter will discuss in detail the background literature related to this thesis. In particular, Section 2.1 surveys the algorithms computing maximal and convex layers whose worst-case running time are bounded asymptotically, including output-sensitive algorithm. Section 2.2 reviews the works concerning the exact constant factors in the bounds on running times. Lastly, Section 2.3 gives an overview for the community's interest in the complexity of geometric structures.

2.1 Algorithms computing maximal and convex layers

Back in the 1970s, Kung et al. [31] presented algorithms to compute the skyline of a point set in 2D or 3D in $O(n \log n)$ worst-case time, which is asymptotically optimal. For a fixed dimension $d \geq 4$, the cost is $O(n \log^{d-2} n)$, which is still the most efficient known solution. Blunck and Vahrenhold [5] introduced in-place algorithms for computing the skyline, which is the first maximal layer, in 2D and 3D, where in-place means that besides the representation of the input, only $O(1)$ extra space is used by the algorithms.

For convex hulls, there are also many classical results. The computational geometry textbook by Berg et al. [4] presents several intuitive and easy-to-implement algorithms to construct the convex hull of a 2D or 3D point set in worst-case optimal time, which is $O(n \log n)$. Surprisingly, the convex hull in high dimensional space, where $d \geq 4$, is not easy to compute. The reason arises from the fact that the complexity of the convex hull in high dimensions matches the number of facets, which may be up to $\Theta(n^{\lfloor d/2 \rfloor})$ in the worst case, rather than the number of vertices as in low dimensions. Starting with an optimal algorithm for only even dimensions in [41], it took a decade before Chazelle and Bernard [13] finally came up with the best worst-case solution in $\Theta(n^{\lfloor d/2 \rfloor})$ time for an arbitrary dimension. Because of the high cost of computing convex hulls in high dimensions, there have been attempts to compute only the extreme points, which are the input points staying on the boundary of the convex hull, and there are only $O(n)$ of them. Golin and Sedgewick

[24] computed a superset of extreme points with size $O(\sqrt{n})$ in $O(n)$ time. For random sets of points sampled from a CI distribution, Bentley et al.'s algorithm [3] returned a superset of extreme points with expected size $O(\log^{d-1} n)$ and used only $2dn + o(n)$ scalar comparisons in expectation.

Computing *multiple* layers is also interesting. *All* maximal layers can be computed in-place in $O(n \log n)$ time for a 2D point set [5]. In 3D, using dynamic fractional cascading, Agarwal [37] presented a solution to compute all maximal layers in $O(n \log n \log\log n)$ time. Later, Buchsbaum and Goodrich [7] improved Agarwal's result to use only $O(n \log n)$ time and $O(n \log n / \log\log n)$ space. For $d \geq 4$, besides the trivial, non-efficient solution of applying Kung et al. [31] recursively, there have been no publications on the problem of computing multiple maximal layers. Researchers have also considered the computation of multiple convex layers. *All* convex layers of a point set can be computed using $O(n \log n)$ time and $O(n)$ space in 2D [12]. For $d \geq 3$, no optimal algorithm for computing multiple convex layers is known.

Along with worst-case algorithms, output-sensitivity of the solutions has also been extensively studied. The classical result from Kirkpatrick and Seidel [28] presented output-sensitive algorithms to compute the skyline in 2D and 3D in $O(n \log h)$ time; and in higher dimensions when $d \geq 4$ in $O(n \log^{d-2} h)$ time, where h is the output size. The first optimal output-sensitive algorithm which computes a convex hull in 2D in $O(n \log h)$ time is credited to Kirkpatrick and Seidel [29]. Later, Chan et al. [10] and Wenger [44] independently simplified [29]'s work, while Chan [9] provided a different, more elegant algorithm. In 3D, at first, Clarkson and Shor [17] and Clarkson [16] presented two optimal, randomized, and output-sensitive convex-hull algorithms. Subsequently, Chazelle and Matoušek [14] and Chan [9] gave two deterministic versions of the algorithms. On the other hand, there is not much work on output-sensitive algorithms for multiple layers yet. Nielsen [35] is the first to present an output-sensitive algorithm for finding the first k convex or maximal layers of a 2D point set in $O(n \log h_k)$ time, where h_k is the number of points in these layers.

2.2 Constant factors in the number of comparisons

Minimizing the constant factors in the leading term of the number of comparisons is common in the mathematics as well as in the computer science literature. In this section, we will focus on computational geometry context and particularly the skyline and convex hull. One of the most well-known results is in [42] for the 2D point location problem, which is to report the face of a planar subdivision containing a given query point. Seidel and Adamy [42] even showed a nice bound on the second term by answering a 2D point location query in $\log n + 2\sqrt{\log n} + O(\log^{1/4} n)$ comparisons, which are point and line comparisons, with linear space. Inspired by [42], Chan and Lee [11] presented an output-sensitive 2D-maximal algorithm which uses only $n \log h + O(n\sqrt{\log h})$ scalar comparisons where h is the output size. In 3D, the randomized algorithm in [11] used at most $n \log h + O(n \log^{2/3} h)$ scalar comparisons in expectation. The same bounds on the number of scalar comparisons are applied to the convex hull in 2D and 3D in [11]. However, since computing the convex hull is a non-orthogonal problem, the definition of a comparison includes both scalar comparisons and *predicate* operations, for example, testing whether three points are in clockwise order or not. It is controversial to show the exact relationship between scalar comparisons and *predicate* operations.

For inputs drawn from the uniform distribution inside a square, Clarkson [16] introduced an algorithm to compute the skyline in $2n + O(n^{5/8})$ scalar comparisons in expectation. Extending the distribution to be any CI distribution, Bentley et al. [3] provided a different algorithm to find the 2D skyline in $2n+o(n)$ scalar comparisons in expectation. Later, Golin [25] showed that another algorithm, suggested but not analyzed by Bentley et al. [3], also achieved the same expected bound on the number of scalar comparisons. The authors of [3] provided algorithms to compute skylines in any d-dimensional space, $d \geq 2$, using at most $dn + o(n)$ scalar comparisons in expectation. Bentley et al. [3] computed the convex hull in $2dn + o(n)$ expected scalar comparisons where the number of dimensions, d, is 2 or 3. The key ingredient of Bentley et al.'s skyline and convex hull algorithms is the ability to prune most of the points in the original set, and these points are not on the skyline or the convex hull. The subset of remaining points is small enough to apply any optimal worst-case algorithm to extract the layer. Such a small subset can only exist if the skyline or convex hull is small, which motivates the need to study the complexity of these layers.

2.3 Complexity of geometric structure

Raynaud [39] proved that if input points are uniformly and independently picked inside a disk, then the expected size of the convex hull is $\Theta(n^{1/3})$. On the other hand, point sets with the same restriction picked from a square will result in an expected $\Theta(\log n)$-size convex hull [40]. Rényi and Sulanke [40] showed that if inputs are randomly and independently picked inside a 2D convex polygon with k sides, then the expected size of the convex hull is $O(k \log n)$. Those results are often included in geometry textbooks, such as [43] and [38]. Bentley et al. [2] extended the results to high dimensions and the skyline as well. They proved that the expected size of the skyline or the expected number of vertices of the convex hull over an arbitrary CI distribution is $O(\log^{d-1} n)$. Sariel [27] later provided different proofs for the same bounds as in [2].

On the "real" complexity of the convex hull in high dimensions, Raynaud [39] showed that the expected number of facets of the convex hull is $O(n^{(d-1)/(d+1)})$ for points sampled from the uniform distribution inside a d-dimensional sphere. In case of points sampled from a d-dimensional normal distribution, the bound changed to $O(\log^{(d-1)/2} n)$ [39]. Besides the expected bound $2\sqrt{n}$ on the number of maximal layers in 2D, Dalal [19] is the first to provide a tight bound in the same manner for convex hull. Dalal [19] showed that the expected number of convex layers is $\Theta(n^{2/(d+1)})$ for a set of n points independently chosen from a uniform distribution inside any bounded, non-empty region in $\mathbb{R}^d$. Readers can refer to Okabe et al. [36] for a review of many other problems in this area.

Chapter 3

Algorithms for maximal and convex layers

This chapter proves our first set of results. At first, Section 3.1 provides an overview of our algorithm. Then Section 3.2 provides and analyzes the algorithms to compute maximal layers, while Section 3.3 discusses the algorithms to compute convex layers in small dimensional space when $d = 2$ or 3. Section 3.4 ends the chapter with the algorithm to compute convex layers in high dimension, $d \geq 4$.

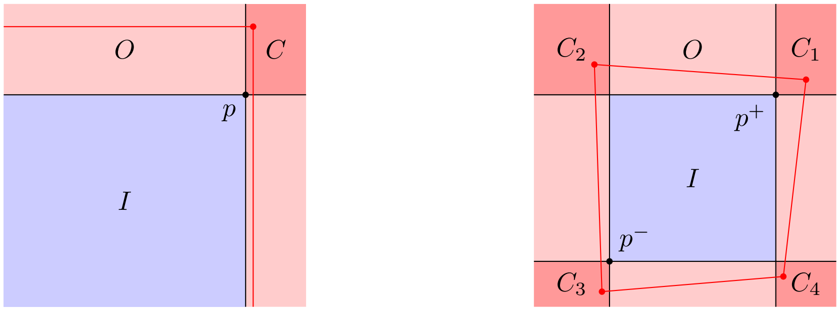
3.1 Algorithm Overview

We start the algorithm outline with Subsection 3.1.1 describing the original idea from Bentley et al. [3]. Subsection 3.1.2, then, gives an overview on how we adapt to compute multiple layers.

3.1.1 The original algorithm to compute one layer

This thesis starts with the following simple idea from Bentley et al.'s paper [3]: Find a rectangular *inner region* I that is expected to contain almost all points in S and is likely to be completely below the skyline of S or inside the convex hull of S (refer to Figure 3.1). In particular, with high probability, the *outer region* $O = \mathbb{R}^d \setminus I$ will contain all points in S that belong to the skyline or convex hull. Surprisingly, as shown in [3], that is enough to compute the skyline or convex hull of a point set S using only $dn + o(n)$ or $2dn + o(n)$ expected comparisons.

Following the definitions of region I and O, the algorithm partitions the point set S into two subsets $S_I = S \cap I$ and $S_O = S \cap O$. Since S_I will likely contain almost all points in S, S_O is small in expectation. Thus, Bentley et al. [3] only used $o(n)$ expected comparisons to compute the skyline, or convex hull, L_O of S_O by some standard skyline or convex hull algorithm. The final step is to check the validity of the calculated layer L_O. This step is done by running a test to check whether certain subregions of O (C and $C_1, \ldots, C_4$, respectively, in Figure 3.1) contain at least one point of S_O. If so, the algorithm terminates in this case.



(a) Maximal layers

(b) Convex layers

Figure 3.1: The inner and outer regions used in Bentley et al.'s and our algorithm illustrated for the 2D case. I is shaded blue. O is shaded pink, including the darker regions, which are the corners that are tested by the algorithm to find out whether they each contain a point. As illustrated in red, any point in C dominates I in the case of maximal layers; in the case of convex layers, the convex hull of any four points in $C_1, \ldots, C_4$ encloses I.

Because I is entirely below or inside L_O, no point in $S \setminus S_O = S_I \subseteq I$ can be on the first maximal layer or convex hull of S. Thus, L_O is also the skyline or convex hull of S. On the other hand, if L_O fails the test, the first maximal or convex layer is extracted by running a standard skyline or convex hull algorithm on S, instead of S_O. Because the cost for this case is too high, I is specifically designed to minimize the probability that this happens. The expected cost of the final step for both cases is, therefore, $o(n)$.

For the skyline algorithm, $I = (-\infty, x_1(p)] \times (-\infty, x_2(p)] \times \cdots \times (-\infty, x_d(p)]$, where p is an appropriate point and $x_i(p)$ denotes the ith coordinate of p. For the convex hull algorithm, $I = [x_1(p^-), x_1(p^+)] \times [x_2(p^-), x_2(p^+)] \times \cdots \times [x_d(p^-), x_d(p^+)]$ for an appropriate pair of corner points (p^-, p^+). In the case that points are distributed uniformly at random in the unit hypercube, the corner points of I can be found without any comparisons by setting $x_i(p^-) = \epsilon$ and $x_i(p) = x_i(p^+) = 1 - \epsilon$ for all $1 \leq i \leq d$ and some appropriate value $\epsilon > 0$. To partition S into S_I and S_O, we compare each point in S to p (or to each point in the pair (p^-, p^+)), which takes d (or $2d$) scalar comparisons. Thus, in total, it takes dn or $2dn$ scalar comparisons for the whole set S.

For an arbitrary CI distribution, the process is slightly different. Specifically, each of p, p^-, and p^+ can be found using $dn + o(n)$ scalar comparisons by randomized linear-time selection. The partitioning of S into S_I and S_O is performed as part of the selection process without incurring any additional comparisons. We discuss this in more detail as part of our high-probability analysis and extension to multiple layers (Lemmas 3.1 and 3.7). Overall, the algorithm takes $dn + o(n)$ or $2dn + o(n)$ expected comparisons, involving in the following procedure: finding p or the pair (p^-, p^+), computing the partition of S into S_I and S_O, and a process for computing and validating the maximal layer or convex hull using $o(n)$ scalar comparisons expectedly.

3.1.2 An overview of our algorithm to compute multiple layers

To extend Bentley et al.'s result to multiple maximal or convex layers, the inner region I needs to contain, again, almost all points in S while the first k layers are unlikely to intersect I. We will modify the choices for the point p or the pair of points (p^-, p^+) to maximize the number of extracted layers. The algorithm starts with computing the corner points of I and the partition of S. Then, the first k layers are obtained from S_O using some known algorithms which have running times bounded in the worst case. The validity of each layer can be tested by checking the existence of input points inside each of certain subregions after removing all computed layers from S. If the test fails, we throw away the results and recompute the first k layers from S.

To achieve the same bound of $dn+o(n)$ or $2dn+o(n)$ on the number of scalar comparisons as [3] and to show that there are high probability results, we need to strengthen the analysis of Bentley et al. to show that (a) computing p or the pair (p^-, p^+) and the partition of S can be done using only $dn + o(n)$ or $2dn + o(n)$ scalar comparisons with high probability and (b) with high probability, no point in I is part of the first k layers.

Since the proofs are slightly simpler, we present our result for maximal layers first. Then, in Section 3.3, we argue that the same approach, with minor modifications, can also be used to compute convex layers.

3.2 Maximal Layers

In this thesis, we use the notation $p \nearrow q$ to mean that point q dominates point p. For any positive value $\tau \in [0, 1]$, point p is a τ-pivot of S if the expected number of points $p' \in S$ satisfying $x_i(p') \geq x_i(p)$ is τn, $\forall i \in [1, d]$, where $x_i(p)$ is p's ith coordinate. Point p is not necessarily in S. To simplify the analysis, we denote $SL^{(k)}(S)$ as the set of the first k maximal layers of the set S. In the case of the first layer, we sometimes omit the superscript and only call $SL(S)$ for convenience. With the definition of I as in Section 3.1, an appropriate τ-pivot p establishes the inner region I. We first prove the following lemma on locating a τ-pivot in general.

Lemma 3.1. *Let S be a point set drawn from a CI distribution. For any value $t > 0$, any value $\tau \in (0, n^{-t}]$, and any constant $\gamma \in (0,1)$, a τ-pivot p and a partition of S into two subsets $S_I = S \cap I$ and $S_O = S \cap (\mathbb{R}^d \setminus I)$ can be computed using $dn + o(n)$ scalar comparisons in expectation and with probability at least $1 - o\left(n^{-n^{\gamma}}\right)$, where I is the region dominated by p.*

Proof. If S is drawn uniformly at random from the unit hypercube, we define $p = (1 - \tau, \ldots, 1 - \tau)$. In the ith dimension, $1 \leq i \leq d$, let G_i be the interval containing all coordinates greater than $x_i(p)$. Then, the length of the interval G_i is τ. Since n points in S are randomly chosen, there are τn expected points in G_i for any dimension. Thus, p is a τ-pivot of S and apparently, p can be found without any comparisons. I is the region containing all points dominated by p. Thus, to partition S into S_I and S_O, we check for each point whether it is dominated by p (and thus belongs to S_I) or not (and thus belongs to S_O). This partitioning process will take d scalar comparisons for each point or dn scalar comparisons in total for the whole set S, that is, the lemma holds in the worst case for a uniform random distribution.

For an arbitrary CI distribution, we want $x_i(p)$ to be the (τn)th largest coordinate in dimension i among the points in S. If so, by definition, p is a τ-pivot. As we show in Appendix A, the (τn)th largest element, among a set of n coordinates in any dimension, can be found using $n + o(n)$ scalar comparisons in expectation and with probability at least $1 - o\left(n^{-n^{\gamma}}\right)$ using a simplified version of LazySelect [34]. While computing p, each coordinate, and therefore the corresponding point in S, is tagged as less than or equal to, or greater than $x_i(p)$. Computing $x_i(p)$ in all d dimensions will give p and take $dn + o(n)$ scalar comparisons in expectation and with probability at least $1 - o\left(n^{-n^{\gamma}}\right)$. The subset S_I is then obtained by gathering all not-greater-than tagged points, while S_O is the complement of S_I from S. This partitioning process requires no further scalar comparisons. □

We now find the requirements for τ to create the right inner region I. Recall from Section 3.1, I needs to likely contain almost all points from S while unlikely intersect with

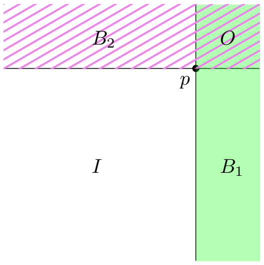


Figure 3.2: Illustration of half-spaces in 2D. B_1 is shaded green while B_2 is filled with pink dashes.

any of the first k maximal layers. Because it is easier for us to analyze from the perspective of the outer region O , we will restate these requirements in an equivalent form: O needs to likely contain very few points from S while enclosing all first k maximal layers completely. As shown later, these two restrictions provide the domain for τ, from which we shall pick the optimal value. We first work with the condition limiting the size of S_O.

Lemma 3.2. *Let S be a point set drawn from a CI distribution, let $0 < \epsilon_1 < \epsilon_2 < 1$ be constants, let $\tau \geq n^{\epsilon_2 - 1}$, let p be a τ-pivot of S, let $S_I \subseteq S$ be the set of points dominated by p, and let $S_O = S \setminus S_I$. Then $E(|S_O|) \leq d\tau n$ and $P(|S_O| > 2d\tau n) = o\big(n^{-n^{\epsilon_1}}\big)$.*

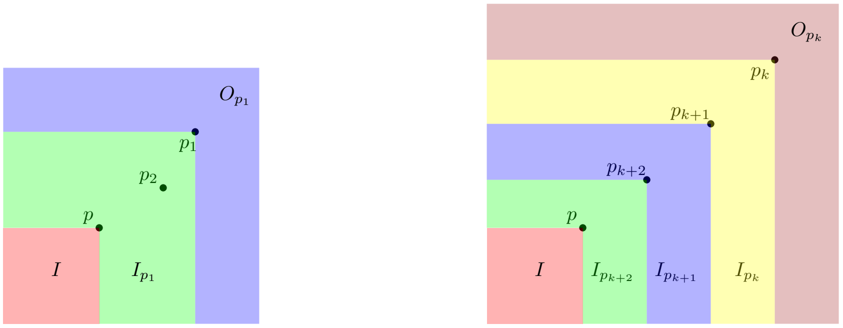
Proof. We can cover the outer region O with d halfspaces $B_1, B_2, \ldots, B_d$, where $B_i = \{p' \in \mathbb{R}^d \mid x_i(p') \geq x_i(p)\}$, refer to Figure 3.2.

From the definition of the τ-pivot p, we have $E(|B_i \cap S|) = \tau n$ and $E(|S_O|) \leq \sum_{i=1}^{d} E(|B_i \cap S|) = d\tau n$. Since $|B_i \cap S|$ is the sum of independent Bernoulli random variables, the Chernoff bound states that

$$Pr(|B_i \cap S| > 2\tau n) < e^{-\tau n/3} \leq n^{-n^{\epsilon_2}/(3 \ln n)} = o\big(n^{-n^{\epsilon_1}}\big).$$

Thus, $P(|S_O| > 2d\tau n) \leq \sum_{i=1}^{d} P(|B_i \cap S| > 2\tau n) = o\big(dn^{-n^{\epsilon_1}}\big) = o\big(n^{-n^{\epsilon_1}}\big)$. □

Now we can focus on analyzing the likelihood for the first k maximal layers to stay completely in O. We first state a condition for the existence of multiple layers, then, integrate the probability to complete the analysis. For the first step, we show a condition for the presence of the first k maximal layers by induction on k in Lemma 3.3. To state this lemma, for an arbitrary point $q \in \mathbb{R}^d$, we define $I_q = (-\infty, x_1(q)] \times (-\infty, x_2(q)] \times \cdots \times (-\infty, x_d(q)]$ and $O_q = \mathbb{R}^d \setminus I_q$.



(a) Base case

(b) Inductive step

Figure 3.3: Visualizing the regions in 2D.

Lemma 3.3. *Let p be an arbitrary point in $\mathbb{R}^d$ and consider the corresponding partition of S into subsets S_I and S_O. If there exist $k+1$ points $p_1, p_2, \ldots, p_{k+1}$ in S_O such that $p \nearrow p_{k+1} \nearrow \cdots \nearrow p_1$, then $SL^{(k)}(S_O) = SL^{(k)}(S)$ and no point in $S \cap I_{p_k}$, except possibly p_k itself, is part of these layers. In particular, $p_{k+1} \notin SL^{(k)}(S)$.*

Proof. The base case is when $k = 1$; refer to Figure 3.3. Since $p_1 \in S_O$ and p_1 dominates all points in $S_O \cap I_{p_1}$, $SL(O_{p_1} \cap S_O) = SL(S_O)$. Since $S_O \subseteq S$, p_1 is also in S. Then, because p_1 dominates all points in $S \cap I_{p_1}$, $SL(O_{p_1} \cap S) = SL(S)$. Since $p \nearrow p_1$, $O_{p_1} \subseteq O$. Therefore, we have $O_{p_1} \cap S_O = O_{p_1} \cap (O \cap S) = (O_{p_1} \cap O) \cap S = O_{p_1} \cap S$. Thus, $SL(S_O) = SL(O_{p_1} \cap S) = SL(S)$. No point in $S \cap I_{p_1}$ is on $SL(S)$ because they are all dominated by point $p_1 \in S$. Since $p_2 \nearrow p_1$, $p_2 \in S \cap I_{p_1}$ and therefore $p_2 \notin SL(S)$.

Suppose that the lemma holds for some integer $k \geq 1$. We will prove that it also applies for $k+1$; refer to Figure 3.3. Indeed, by the inductive hypothesis, $SL^{(k)}(O_{p_{k+2}} \cap S) = SL^{(k)}(S)$ and $SL^{(k)}(O_{p_{k+2}} \cap S_O) = SL^{(k)}(S_O)$, when we replace S by S_O in the lemma. Since $p \nearrow p_{k+2}$, $O_{p_{k+2}} \subseteq O$. Therefore, we have $O_{p_{k+2}} \cap S_O = O_{p_{k+2}} \cap (O \cap S) = (O_{p_{k+2}} \cap O) \cap S = O_{p_{k+2}} \cap S$. Thus, $SL^{(k)}(S_O) = SL^{(k)}(O_{p_{k+2}} \cap S) = SL^{(k)}(S)$ and no point in $S \cap I_{p_k}$ is part of these layers.

Now, we define $S' = S \setminus SL^{(k)}(S)$ as the set left after removing the first k maximal layers from the original. By definition, $SL(S')$ is also the $(k+1)$st maximal layer of S. Because p_{k+1} and p_{k+2} are in $S \cap I_{p_k}$, p_{k+2} and p_{k+1} are in S'. Applying the base case for set S' and $p \nearrow p_{k+2} \nearrow p_{k+1}$, we get $SL(S' \cap O) = SL(S')$ and no point in $S \cap I_{p_{k+1}}$ is part of this layer. Because $SL^{(k)}(S_O) = SL^{(k)}(S)$, $SL(S' \cap O)$ is also the $(k+1)$st maximal layer of S_O. Combining all pieces together, we have $SL^{(k+1)}(S_O) = SL^{(k+1)}(S)$ and no point in $S \cap I_{p_{k+1}}$ is part of these layers. In particular, $p_{k+2} \nearrow p_{k+1}$, $p_{k+2} \in S \cap I_{p_{k+1}}$ and therefore, $p_{k+2} \notin SL^{(k+1)}(S)$. □

Finally, we provide the probability for $k+1$ points, p_1 to p_{k+1}, in Lemma 3.3 to exist.

Lemma 3.4. *Let S be a point set drawn from a CI distribution, let $0 < \epsilon_1 < \epsilon_2 < 1$ be constants, let $\frac{\tau}{k+1} \geq n^{(\epsilon_2-1)/d}$, and let $h_0, h_1, \ldots, h_{k+1}$ be $k+2$ points such that h_j is a $\left(\frac{j}{k+1}\tau\right)$-pivot of S for all $0 \leq j \leq k+1$. Then with probability at least $1 - o\left(n^{-n^{\epsilon_1}}\right)$, each hyperrectangle H_j defined by points h_{j-1} and h_j, for $1 \leq j \leq k+1$, contains a point $p_j \in S$. These points satisfy $p_{k+1} \nearrow p_k \nearrow \cdots \nearrow p_1$.*

Proof. Consider an arbitrary hyperrectangle H_j, see Figure 3.4. Since h_{j-1} and h_j are the $\left(\frac{j-1}{k+1}\tau\right)$-pivot and $\left(\frac{j}{k+1}\tau\right)$-pivot correspondingly, $Pr(x_i(p) \in [x_i(h_j), x_i(h_{j-1})]) = \frac{\tau}{k+1}$, $\forall p \in S$, $\forall i \in [1,d]$. Because the coordinates are chosen independently, $Pr(p \in H_j) = \left(\frac{\tau}{k+1}\right)^d$. Thus, $E(|H_j \cap S|) = \left(\frac{\tau}{k+1}\right)^d n$. Since $|H_j \cap S|$ is the sum of independent Bernoulli random variables, the Chernoff bound states that

$$P(H_j \cap S = \emptyset) < e^{-(\tau/(k+1))^d n/4}$$

As there are $k+1$ H_j's, the probability that there exists at least one hyperrectangle containing no input point in its interior is at most

$$(k+1)e^{-(\tau/(k+1))^d n/4} \leq (k+1)e^{-n^{\epsilon_2}/4} \leq n^{1-n^{\epsilon_2}/(4\ln n)} = o\left(n^{-n^{\epsilon_1}}\right)$$

for $\frac{\tau}{k+1} \geq n^{(\epsilon_2-1)/d}$, $\epsilon_1 < \epsilon_2$, and $k+1 \leq n$. □

We are now ready to prove Theorem 3.5, which transforms an algorithm computing multiple layers in worst-case $O(k^c n^{1+\epsilon})$ scalar comparisons into an algorithm computing the same number of maximal, but using only $dn + o(n)$ scalar comparisons, in expectation and also with high probability.

Theorem 3.5. *Let S be a set of n points drawn from an arbitrary CI distribution in d dimensions. Suppose there is an algorithm M that can compute the first k maximal layers of S using $O(k^c n^{1+\epsilon})$ scalar comparisons in the worst case, where c and ϵ are constants with*

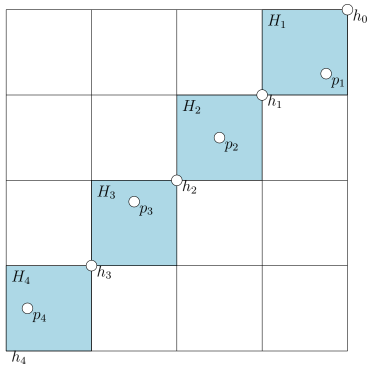


Figure 3.4: An example of defining H_i's in 2D, where $k = 3$. In this figure, the entire big rectangle is C and each H_i represents a blue shaded square. It also shows the points p_i's defined in Lemma 3.4.

$c \geq 0$ and $0 < \epsilon < \frac{1}{(c+1)d}$. Then the first $\kappa = n^{\frac{1}{(c+1)d}-\epsilon}$ maximal layers of S can be computed using $dn + o(n)$ expected scalar comparisons, and the actual number of comparisons is within the same bounds with probability $1 - o(n^{-n^{\gamma}})$ for any $\gamma \in (0, (c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)})d)$.

Proof. Our algorithm starts by applying Lemma 3.1 to find a τ-pivot p and the partition of S into S_I and S_O (the valid range for τ is shown at the end of this proof). It, then, computes the first k maximal layers of S_O using M. And to finish, the algorithm performs a check to see whether there exists a point in S_O dominating p, after removing the computed maximal layers. If it passes the test, then no point in S_I can contribute to $SL^{(k)}(S)$. In other words, $SL^{(k)}(S_O) = SL^{(k)}(S)$, so the algorithm reports the computed maximal layers. Otherwise, it runs M on S to compute $SL^{(k)}(S)$.

What left is to analyze the running time. We are only interested in the number of scalar comparisons. We prove that this algorithm uses $dn + o(n)$ scalar comparisons with high probability. The total number of comparisons in the algorithm is $dn + o(n)$ if (a) computing p and partitioning S into S_I and S_O takes $dn + o(n)$ comparisons, (b) running algorithm M on S_O incurs $o(n)$ comparisons, and (c) the algorithm passes the final test, which means there exists a point in S_O not on the first k maximal layers while dominating p. Thus, it suffices to bound the probability that any of these three conditions fails.

By Lemma 3.1, (a) fails with probability $o\Big(n^{-n^{\gamma}}\Big)$, for any $\gamma \in (0, 1)$, as long as $\tau = n^{-t}$ for some $t > 0$. Running algorithm M on S_O incurs $o(n)$ scalar comparisons if $|S_O| =$

$o\big(n^{1/(1+\epsilon)}\big)/k^c$. By Lemma 3.2, $|S_O| \le 2d\tau n$ with probability $1 - o\Big(n^{-n^{\gamma}}\Big)$ as long as $\tau \ge n^{\epsilon_2 - 1}$ for some $\epsilon_2 > \gamma$. Therefore, (b) fails with probability $o\Big(n^{-n^{\gamma}}\Big)$ as long as $\tau n = o\big(n^{1/(1+\epsilon)}\big)/k^c$ and $\tau \ge n^{\epsilon_2 - 1}$. By Lemma 3.3 and Lemma 3.4, (c) fails with probability $o\Big(n^{-n^{\gamma}}\Big)$ as long as $\frac{\tau}{k+1} \ge n^{(\epsilon_2 - 1)/d}$ for some $\epsilon_2 > \gamma$. Thus, the probability that any of these three conditions fails is $o(n^{-n^{\gamma}})$, if τ is in a *valid domain* to hold the above constraints.

First observe that $\epsilon_2 - 1 < 0$. Thus, $\tau \ge n^{\epsilon_2 - 1}$ if $\frac{\tau}{k+1} \ge n^{(\epsilon_2 - 1)/d}$, so it suffices to choose a value of $\tau = n^{-t}$, for some $t > 0$, such that $\frac{\tau}{k+1} \ge n^{(\epsilon_2 - 1)/d}$ and $\tau n = o\big(n^{1/(1+\epsilon)}\big)/k^c$. The last two constraints imply that $k^{c+1} = o(n^{-\epsilon/(\epsilon+1)+(1-\epsilon_2)/d})$ or $k = o(n^{\frac{1}{(c+1)d} - \epsilon + \delta})$ where $\delta = \epsilon - \frac{\epsilon_2}{(c+1)d} - \frac{\epsilon}{(\epsilon+1)(c+1)}$. For any $\epsilon_2 < \big(c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)}\big)d$, we have $\delta > 0$, that is, we can compute up to $n^{\frac{1}{(c+1)d} - \epsilon}$ maximal layers and, since $(c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)}) > 0$, we can choose values γ and ϵ_2 such that $0 < \gamma < \epsilon_2 < (c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)})d$.

It remains to choose τ. We have $\tau n = o\big(n^{1/(1+\epsilon)}\big)/k^c$ if $t > \frac{\epsilon}{1+\epsilon} + \frac{c}{(c+1)d} - \epsilon c$. To satisfy $\frac{\tau}{k+1} \ge n^{(\epsilon_2 - 1)/d}$ for large enough n, we need $t < \epsilon - \frac{1}{(c+1)d} - \frac{\epsilon_2 - 1}{d}$, which is true as long as $t < \epsilon - \frac{1}{(c+1)d} - \epsilon c - \frac{\epsilon^2}{2(1+\epsilon)} + \frac{1}{d}$ because $\epsilon_2 < (c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)})d$. It is easy to verify that $\frac{\epsilon}{1+\epsilon} + \frac{c}{(c+1)d} - \epsilon c < \epsilon - \frac{1}{(c+1)d} - \epsilon c - \frac{\epsilon^2}{2(1+\epsilon)} + \frac{1}{d}$. Thus, we can choose a value of t that satisfies both constraints and set $\tau = n^{-t}$. In addition, since $\epsilon < \frac{1}{(c+1)d}$, we have $\frac{\epsilon}{1+\epsilon} + \frac{c}{(c+1)d} - \epsilon c > 0$, that is, $t > 0$.

With the chosen value of τ, we analyze the expected bound on the number of scalar comparisons in our algorithm. Our algorithm can be viewed as a sequence of three actions: (1) finding p and corresponding partition of S, (2) applying M to S_O and (3) either checking the condition or applying M to S. Let X, Y, Z correspondingly be the number of scalar comparisons used in each action. The expected number of scalar comparisons used by our algorithm is $E(X + Y + Z) = E(X) + E(Y) + E(Z)$. By Lemma 3.1, $E(X) = dn + o(n)$. By Lemma 3.2, we have

$$\begin{aligned} E(Y) &\le O\big(k^c(2d\tau n)^{1+\epsilon}\big)Pr(|S_O| \le 2d\tau n) + O(k^c n^{1+\epsilon})o\big(n^{-n^{\gamma}}\big) \\ &< O\big((k^c\tau n)^{1+\epsilon}\big) + o(n) = o(n) \end{aligned}$$

where $d = O(1), t > 0$ and $\tau n k^c = o\big(n^{1/(1+\epsilon)}\big)$.

To compute $E(Z)$, observe that action (3) could be split into three cases: (c1) passing the test and $S_O \le 2d\tau n$, (c2) passing the test and $S_O > 2d\tau n$, (c3) failing the test. In any cases, our algorithm performs $d|S_O|$ scalar comparisons for the test. Let Z_i, for $i \in \{1, 2, 3\}$, be the number of scalar comparisons used in each corresponding case. Suppose each case can occur with probability $Pr(Z_i)$ correspondingly. By Lemma 3.2, $Pr(Z_2) = o\big(n^{-n^{\gamma}}\big)$. By Lemma 3.3 and Lemma 3.4, $Pr(Z_3) = o\Big(n^{-n^{\gamma}}\Big)$. Thus,

$$\begin{aligned}
Z_1 Pr(Z_1) &\leq 2d^2 \tau n = o(n) \\
Z_2 Pr(Z_2) &\leq dno\big(n^{-n^{\gamma}}\big) = o(n) \\
Z_3 Pr(Z_3) &\leq (dn + O(k^c n^{1+\epsilon}))o\big(n^{-n^{\gamma}}\big) = o(n)
\end{aligned}$$

where $d = O(1), t > 0$ and $\tau n k^c = o\big(n^{1/(1+\epsilon)}\big)$. It implies that $E(Z) = \sum_{i=1}^{3} Z_i Pr(Z_i) = o(n)$. Finally, we conclude that $E(X + Y + Z) = dn + o(n)$. □

We then apply Theorem 3.5 to achieve the following results on computing maximal layers:

Theorem 3.6. *Let S be a set of n points drawn from an arbitrary CI distribution in d dimensions. If $d \in \{2, 3\}$ and $\epsilon > 0$, the first $n^{\frac{1}{d}-\epsilon}$ maximal layers of S can be computed using $dn + o(n)$ expected scalar comparisons. If $d \geq 4$, the first $n^{\frac{1}{2d}-\epsilon}$ maximal layers can be computed using the same expected number of scalar comparisons. In all these cases, the actual number of comparisons is within the same upper bounds on the expected number of comparisons with probability $1 - o(n^{-n^{\gamma}})$, where γ is as in Theorem 3.5.*

Proof. Since all maximal layers can be computed in $O(n \log n)$ time in 2D by [5] and in 3D by [7], there exist algorithms M to compute the first k 2D, 3D-maximal layers using $O(k^c n^{1+\epsilon})$ scalar comparisons in the worst case, where $c = 0$ and $0 < \epsilon < \frac{1}{(c+1)d}$. For $d \geq 4$, Kung et al.'s algorithm [31] computes one maximal layer in $O(n \log^{d-2} n)$ time, which is $O(n^{1+\epsilon})$ for any $0 < \epsilon$. Thus, we can extend the algorithm in [31] to compute the first k layers in $O(k^c n^{1+\epsilon})$ worst-case time (and therefore the number of scalar comparisons), where $c = 1$. Therefore, we can apply Theorem 3.5 to give the conclusion. □

3.3 Convex Layers in Two and Three Dimensions

To extend the framework from Section 3.2 to compute convex layers, we introduce the notion of dominance to the 2^d possible quadrants of a point in $\mathbb{R}^d$. Each quadrant is identified by a *sign vector* $\sigma \in \{+1, -1\}^d$. A point $q \in \mathbb{R}^d$ *σ-dominates* another point $p \in \mathbb{R}^d$, written as $p \nearrow_\sigma q$, if $\sigma \circ q$ dominates $\sigma \circ p$, where $p \circ q$ is the Hadamard product: $p \circ q = (x_1(p)x_1(q), x_2(p)x_2(q), \ldots, x_d(p)x_d(q))$. For an arbitrary value $\tau \in [0,1]$ and any sign vector σ, point p is a (τ,σ)*-pivot* of S if the expected number of points $p' \in S$ satisfying $x_i(\sigma)x_i(p') \geq x_i(\sigma)x_p(p)$ is τn, for all $i \in [1,d]$. For given values of τ, σ, it is possible to choose multiple (τ,σ)-pivots.

We give particular names to emphasize the importance of these two sign vectors $\mathbf{1} = (1,\ldots,1)$ and $-\mathbf{1} = (-1,\ldots,-1)$. From the definition of σ-dominance, $\mathbf{1}$-dominance is the same as normal dominance and a $(\tau,\mathbf{1})$-pivot is just a τ-pivot in Section 3.2. When $0 < \tau < 1/2$, let p^-, p^+ be one of the possible $(\tau,-\mathbf{1})$-pivots and $(\tau,\mathbf{1})$-pivots correspondingly. As said in Section 3.1, (p^-, p^+) shapes the *inner region* $I = [x_1(p^-), x_1(p^+)] \times [x_2(p^-), x_2(p^+)] \times \cdots \times [x_d(p^-), x_d(p^+)]$ and *outer region* $O = \mathbb{R}^d \setminus I$; see Figure 3.1. It is not hard to see that I contains all points in $\mathbb{R}^d$ that dominate p^- and are dominated by p^+. In term of σ-dominance, I contains all points $\mathbf{1}$-dominated by p^+ and $-\mathbf{1}$-dominated by p^-.

Similar to Section 3.2, S is partitioned into $S_I = S \cap I$ and $S_O = S \cap O$. All 2^d corners of the hyperrectangle I are the points $\{p^\sigma \mid \sigma \in \{+1,-1\}^d\}$, where $p^\sigma = \frac{1}{2}((\mathbf{1}+\sigma)\circ p^+ + (\mathbf{1}-\sigma)\circ p^-)$. Since S is drawn from a CI distribution, each such corner p^σ is also a (τ,σ)-pivot of S. Let $CH^{(k)}(S)$ be the set of the first k convex layers of the set S, where $CH^{(1)}(S)$ is sometimes simplified as $CH(S)$. Again, we need to find appropriate domain for τ to construct the right inner region I. We start with the following partition lemma.

Lemma 3.7. *Let S be a point set drawn from a CI distribution. For any value $t > 0$, any value $\tau \in (0, n^{-t}]$, and any constant $\gamma \in (0,1)$, a pair of points (p^-, p^+) such that p^- is a $(\tau,-\mathbf{1})$-pivot of S and p^+ is a $(\tau,\mathbf{1})$-pivot of S and a partition of S into two subsets $S_I = S\cap I$ and $S_O = S \cap O$ can be computed using $2dn + o(n)$ scalar comparisons in expectation and with probability at least $1 - o\Big(n^{-n^\gamma}\Big)$.*

Proof. Without much work, Lemma 3.1 can be generalized to σ-dominance. In particular, the generalized version of Lemma 3.1 can find p^σ and corresponding partition in $dn + o(n)$ scalar comparisons in expectation and with probability at least $1 - o\Big(n^{-n^\gamma}\Big)$; refer to Figure 3.5. Thus, applying the modified Lemma 3.1 twice with $\mathbf{1}$-dominance and $-\mathbf{1}$-dominance matches the claimed number of scalar comparisons. Let $S_{I_{p^\sigma}}$ be the set of input points σ-dominated by p^σ. Since $S_{I_{p^+}}$ and $S_{I_{p^-}}$ are computed, $S_I = S_{I_{p^+}} \cap S_{I_{p^-}}$ and $S_O = S \setminus S_I$ can be produced

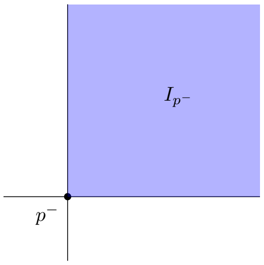


Figure 3.5: Illustration of the generalized version of Lemma 3.1 with $-\mathbf{1}$-dominance in 2D. The inner region I_{p^-} is shaded blue.

without any additional scalar comparisons. □

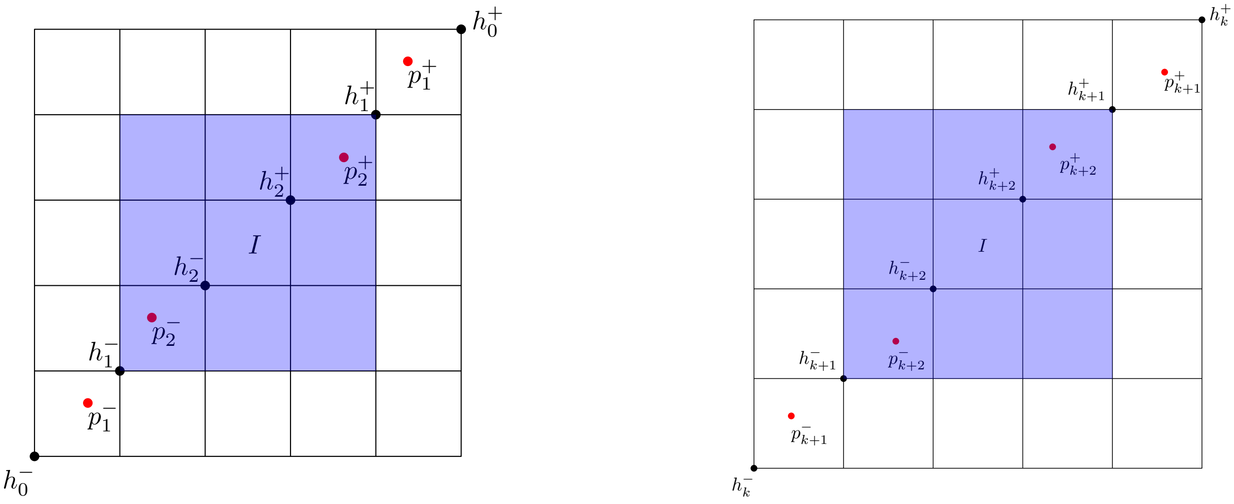
We next analyze the size of S_O.

Lemma 3.8. *Let S be a point set drawn from a CI distribution, let $0 < \epsilon_1 < \epsilon_2 < 1$ be constants, let $\tau \geq n^{\epsilon_2 - 1}$, let p^- be a $(\tau, -\mathbf{1})$-pivot of S, let p^+ be a $(\tau, \mathbf{1})$-pivot of S, let I be the hyperrectangle defined by (p^-, p^+), let $S_I = S \cap I$, and let $S_O = S \setminus S_I$. Then $E(|S_O|) \leq 2d\tau n$ and $P(|S_O| > 4d\tau n) = o\big(n^{-n^{\epsilon_1}}\big)$.*

Proof. The proof is identical to the proof of Lemma 3.2 after observing that S_O can be covered with $2d$ halfspaces $B_1^-, B_2^-, \ldots, B_d^-, B_1^+, B_2^+, \ldots, B_d^+$, where $B_i^- = \{p' \in \mathbb{R}^d \mid x_i(p') \leq x_i(p^-)\}$ and $B_i^+ = \{p' \in \mathbb{R}^d \mid x_i(p') \geq x_i(p^+)\}$ for all $1 \leq i \leq d$. □

We now compute the probability of $CH^{(k)}(S) \subseteq O$. Similar to maximal layers, we also find an indicator for the existence of multiple layers, at first. Then, we find out the probability that this signal may occur. We need the following definitions: given two points h^-, h^+ such that $h^- \nearrow h^+$, the pair (h^-, h^+) defines an axis-aligned hyperrectangle I_h. The hyperrectangle I_h has 2^d corners $\{h^\sigma \mid \sigma \in \{+1, -1\}^d\}$ where $h^\sigma = \frac{1}{2}((\mathbf{1}+\sigma) \circ h^+ + (\mathbf{1}-\sigma) \circ h^-)$. Let $S_{I_h} = S \cap I_h$. We start with the following lemma:

Lemma 3.9. *Let $h_0^-, h_1^-, \ldots, h_{k+1}^-$ and $h_0^+, h_1^+, \ldots, h_{k+1}^+$ be points such that $h_0^- \nearrow h_1^- \nearrow \cdots \nearrow h_{k+1}^- \nearrow h_{k+1}^+ \nearrow h_k^+ \nearrow \cdots \nearrow h_0^+$ and consider the regions I and O defined by the pair of points $p^- = h_{k+1}^-$ and $p^+ = h_{k+1}^+$. If, for every sign vector $\sigma \in \{+1, -1\}^d$, there exist $k+1$ points $p_1^\sigma, p_2^\sigma, \ldots, p_{k+1}^\sigma$ in S_O such that $h_j^\sigma \nearrow^\sigma p_j^\sigma \nearrow^\sigma h_{j-1}^\sigma$ for all $1 \leq j \leq k+1$, then $CH^{(k)}(S_O) = CH^{(k)}(S)$ and no point in $S_{I_{h_k}}$ is part of these layers. In particular, $p_{k+1}^\sigma \notin CH^{(k)}(S)$ for any $\sigma \in \{+1, -1\}^d$.*



(a) Base case

(b) Inductive step

Figure 3.6: Illustration in 2D. The inner regions I_{h_1} and $I_{h_{k+1}}$ are shaded blue. Note that $h_2^+ = p^+$ and $h_2^- = p^-$ in (a) while $h_{k+2}^+ = p^+$ and $h_{k+2}^- = p^-$ in (b).

Proof. In this proof, unless σ is specifically given a value, any statement using the sign vector σ will apply for every $\sigma \in \{+1, -1\}^d$. Since the points all come in pairs, we write $v_i^\sigma \nearrow^\sigma v_j^\sigma$ in short for $v_j^- \nearrow v_i^- \nearrow v_i^+ \nearrow v_j^+$, for any two pairs (v_i^-, v_i^+) and (v_j^-, v_j^+). Let $O_{h_j} = \mathbb{R}^d \setminus I_{h_j}$, for all $0 \leq j \leq k+1$.

The base case is when $k = 1$; refer to Figure 3.6. Since $h_1^\sigma \nearrow^\sigma p_1^\sigma$ and $p_1^\sigma \in S_O$, no point in $S_O \cap I_{h_1}$ can be on $CH(S_O)$. In other words, $CH(O_{h_1} \cap S_O) = CH(S_O)$. Since p_1^σ is also in S, no point in $S \cap I_{h_1}$ can be on $CH(S)$. Thus, $CH(O_{h_1} \cap S) = CH(S)$. Since $p^\sigma \nearrow^\sigma h_1^\sigma$, $O_{h_1} \subseteq O$. Therefore, we have $O_{h_1} \cap S_O = O_{h_1} \cap (O \cap S) = (O_{h_1} \cap O) \cap S = O_{h_1} \cap S$. Thus, $CH(S_O) = CH(O_{h_1} \cap S) = CH(S)$. Since $p_2^\sigma \nearrow h_1^\sigma$, $p_2^\sigma \in I_{h_1} \cap S$ and therefore $p_2^\sigma \notin CH(S)$.

Suppose that the lemma holds for some integer $k \geq 1$. We will prove that it also applies for $k+1$; refer to Figure 3.6. By the inductive hypothesis, $CH^{(k)}(O_{h_{k+1}} \cap S) = CH^{(k)}(S)$ and $CH^{(k)}(O_{h_{k+1}} \cap S_O) = CH^{(k)}(S_O)$, when we replace S by S_O in the lemma. Since $p^\sigma \nearrow^\sigma h_{k+1}^\sigma$, $O_{h_{k+1}} \subseteq O$. Therefore, we have $O_{h_{k+1}} \cap S_O = O_{h_{k+1}} \cap (O \cap S) = (O_{h_{k+1}} \cap O) \cap S = O_{h_{k+1}} \cap S$. Thus, $CH^{(k)}(S_O) = CH^{(k)}(O_{h_{k+1}} \cap S) = CH^{(k)}(S)$ and no point in $S \cap I_{h_k}$ is part of these layers.

Now, we define $S' = S \setminus CH^{(k)}(S)$ as the set left after removing the first k convex layers from the original. By definition, $CH(S')$ is also the $(k+1)$st convex layer of S. Because

p^σ_{k+1} and p^σ_{k+2} are in $S \cap I_{h_k}$, p^σ_{k+2} and p^σ_{k+1} are in S'. Applying the base case for set S' and $h^\sigma_{k+2} \nearrow^\sigma h^\sigma_{k+1} \nearrow^\sigma h^\sigma_k$, we get $CH(S' \cap O) = CH(S')$ and no point in $S \cap I_{h_{k+1}}$ is part of this layer. Because $CH^{(k)}(S_O) = CH^{(k)}(S)$, $CH(S' \cap O)$ is also the $(k+1)$st convex layer of S_O. Combining all pieces together, we have $CH^{(k+1)}(S_O) = CH^{(k+1)}(S)$ and no point in $S \cap I_{h_{k+1}}$ is part of these layers. In particular, since $p^\sigma_{k+2} \nearrow^\sigma h^\sigma_{k+1}$, $p^\sigma_{k+2} \in S \cap I_{h_{k+1}}$ and therefore, $p^\sigma_{k+2} \notin CH^{(k+1)}(S)$. □

Lastly, we provide the probability for the condition.

Lemma 3.10. *Let S be a point set drawn from a CI distribution, let $0 < \epsilon_1 < \epsilon_2 < 1$ be constants, let $\frac{\tau}{k+1} \geq n^{(\epsilon_2 - 1)/d}$, and let $h^-_0, h^-_1, \ldots, h^-_{k+1}, h^+_0, h^+_1, \ldots, h^+_{k+1}$ be points such that h^-_j is a $\left(\frac{j}{k+1}\tau, -\mathbf{1}\right)$-pivot and h^+_j is a $\left(\frac{j}{k+1}\tau, \mathbf{1}\right)$-pivot for all $0 \leq j \leq k+1$. Then with probability at least $1 - o\left(n^{-n^{\epsilon_1}}\right)$, every hyperrectangle H^σ_j defined by the points h^σ_{j-1} and h^σ_j, for $1 \leq j \leq k+1$ and every sign vector $\sigma \in \{+1, -1\}^d$, contains a point $p^\sigma_j \in S$.*

Proof. Analogous to the proof of Lemma 3.4, $P(H^\sigma_j \cap S = \emptyset) < e^{-(\tau/(k+1))^d n/4}$, so the probability that there exists a pair (j, σ) such that $H^\sigma_j \cap S = \emptyset$ is less than $(k+1)2^d e^{-(\tau/(k+1))^d n/4}$. As shown in the proof of Lemma 3.4, $(k+1)e^{-(\tau/(k+1))^d n/4} = o\left(n^{-n^{\epsilon_1}}\right)$. Since d is a constant, this implies that $(k+1)2^d e^{-(\tau/(k+1))^d n/4} = o\left(n^{-n^{\epsilon_1}}\right)$. □

Similar to Section 3.2, we first prove Theorem 3.11 to transform an algorithm computing multiple layers in worst-case $O(k^c n^{1+\epsilon})$ scalar comparisons into an algorithm computing the same number of convex layers, but using only $2dn + o(n)$ scalar comparisons, in expectation and also with high probability.

Theorem 3.11. *Let S be a set of n points drawn from an arbitrary CI distribution in d dimensions. Suppose there is an algorithm M that can compute the first k convex layers of S using $O(k^c n^{1+\epsilon})$ scalar comparisons in the worst case, where c and ϵ are constants with $c \geq 0$ and $0 < \epsilon < \frac{1}{(c+1)d}$. Then the first $\kappa = n^{\frac{1}{(c+1)d} - \epsilon}$ convex layers of S can be computed using $2dn + o(n)$ expected scalar comparisons, and the actual number of comparisons is within the same bounds with probability $1 - o(n^{-n^\gamma})$ for any $\gamma \in (0, (c\epsilon + \frac{\epsilon^2}{2(\epsilon+1)})d)$.*

Proof. Our algorithm picks (p^-, p^+) and partitions set S into S_I and S_O by Lemmas 3.7. It then uses algorithm M to compute the first k convex layers of S_O. Finally, we check whether, for every $\sigma \in \{+1, -1\}^d$, there exists a point σ-dominating p^σ in S_O after the computed convex layers are removed. If this test succeeds, the algorithm reports the computed convex layers. Otherwise, it runs M on S to compute the first k convex layers of S.

The analysis of this algorithm is identical to the proof of Theorem 3.5 for maximal layers using Lemmas 3.7, 3.8, 3.9, and 3.10, in place of Lemmas 3.1, 3.2, 3.3, and 3.4. □

Finally, we prove the results for computing convex layers when $d = 2$ or 3.

Theorem 3.12. *Let S be a set of n points drawn from an arbitrary CI distribution in d dimensions. If $d = 2$, the first $n^{\frac{1}{d}-\epsilon}$ convex layers of S can be computed using $2dn + o(n)$ expected scalar comparisons. If $d = 3$, the first $n^{\frac{1}{2d}-\epsilon}$ convex layers can be computed using this number of expected scalar comparisons. In all these cases, the actual number of comparisons is within the same upper bounds on the expected number of comparisons with probability $1 - o(n^{-n^\gamma})$, where γ is as in Theorem 3.11.*

Proof. The first k convex layers in 2D can be computed using $O(k^c n^{1+\epsilon})$ scalar comparisons where $c = 0$ because all convex layers can be computed as shown in [12] in $O(n \log n)$ worst-case time. For $d = 3$, the $O(n \log n)$-time algorithm in [4] can recursively extract the first k convex layers using $O(k^c n^{1+\epsilon})$ scalar comparisons where $c = 1$ in the worst case. In a nutshell, we just proved that there exists an algorithm M satisfying the condition in Theorem 3.11, for all cases. Thus, we can apply Theorem 3.11 directly to get Theorem 3.12. □

3.4 Convex Layers in Four or Higher Dimensions

This section discusses the problem of computing the first k convex layers of a point set S drawn from an arbitrary CI distribution in four or higher dimensions, for $k \leq n^{1/(d^2+2)}$. The worst-case optimal algorithm for computing the convex *hull* (only one layer) in $d \geq 4$ dimensions [13] takes $O(n^{\lfloor d/2 \rfloor})$ comparisons. $O(n^{\lfloor d/2 \rfloor})$ is so high that if we make S_O small enough to bound the cost, the probability of achieving this cost would asymptotically be close to zero. We need a more efficient way to extract the layers. Fortunately, Bentley et al. [2] already gave us the hint.

Let $\sigma \in \{+1, -1\}^d$ be a sign vector, let $\sigma \circ S = \{\sigma \circ p \mid p \in S\}$, and let L^σ be the set of points $p \in S$ such that $\sigma \circ p$ belongs to the skyline of $\sigma \circ S$. We call L^σ the *σ-skyline* of S. Bentley et al. [2] showed that $Q' = \bigcup_{\sigma \in \{+1,-1\}^d} L^\sigma$ is a superset of extreme points, which are points in S staying on the boundary of $CH(S)$. However, the authors in [2] stopped at computing the superset. Matoušek and Plecháč [33] later called Q' the *quadrant hull* of S and gave a different definition. For an arbitrary point $q \in \mathbb{R}^d$ and a sign vector $\sigma \in \{+1,-1\}^d$, let $ort_\sigma(q) = \{x \in \mathbb{R}^d | sgn(x-q) = \sigma\}$ be the σ-orthant at q and $\overline{ort_\sigma(q)} = \mathbb{R}^d \setminus ort_\sigma(q)$, where $sgn(x)$ is the sign vector of x. Then, the *quadrant hull* of S defined in [33] is $\bigcap\{\overline{ort_\sigma(q)} | S \subseteq \overline{ort_\sigma(q)}, \forall q \in \mathbb{R}^d, \sigma \in \{+1,-1\}^d\}$. We opt for *orthant hull* here because an orthant is the generalization of quadrants to higher dimensions. We will show that the orthant hull definition in [33] is equivalent to a region Z with the points $\bigcup_{\sigma \in \{+1,-1\}^d} L^\sigma$ on its boundary. Indeed, if $S \subseteq \overline{ort_\sigma(q)}$, then q is not σ-dominated by any point in S. It means that, for a given $\sigma \in \{+1,-1\}^d$, L^σ forms a boundary for Z by the skyline definition. Therefore, a point q is in the inner region of Z if and only if $\forall \sigma \in \{+1,-1\}^d, \exists p \in S \mid q \nearrow^\sigma p$. It means that, by definition, no point in $Q' = \bigcup_{\sigma \in \{+1,-1\}^d} L^\sigma$ will stay inside Z. Thus, Q' is the exact set of points in S staying on the boundary of the orthant hull.

With little effort, the construction that proves Theorem 3.11 can be modified to obtain the following theorem:

Theorem 3.13. *Let S be a set of n points in $d \geq 4$ dimensions drawn from an arbitrary CI distribution. For any $k \leq n^{1/(d^2+2)}$, the first k convex layers of S can be found using $2dn + o(n)$ scalar comparisons with probability $1 - O(\frac{1}{n^{1/d-\epsilon}})$, for any $\epsilon > 0$.*

Proof. Applying Kung et al.'s algorithm in [31] to compute all 2^d σ-skylines in $d \geq 4$ dimensions where $\sigma \in \{+1,-1\}^d$, we get set Q' in $O(n \log^{d-2} n)$-time. It is trivial that $CH(Q') = CH(S)$. As shown later, with high probability, the size of Q' is small enough so that applying Chazelle's convex hull algorithm to Q' takes $O(n)$ comparisons. Altogether, this gives an algorithm M to compute $CH^{(1)}(S)$ using $O(n \log^{d-2} n)$ comparisons with high probability. To compute $CH^{(k)}(S)$ in general, our algorithm computes the convex hull of the point set left in the ith iteration after removing the first $i-1$ convex layers. As we show below, with high probability, this will take $O(kn \log^{d-2} n)$ time. Our analysis is based on the fact, proven later, that all supersets of the extreme points used in our algorithm are included in the first k *orthant layers*, which has expected size $O(k^d \log^{d-1} n)$.

Theorem 3.11 requires M to achieve a running time of $O(kn^{1+\epsilon})$ in the worst case. However, if M only attains the same running time with probability at least $p > 0$, we can still modify the proof of Theorem 3.11 to capture probability p in the final result as well. Indeed, similar to Sec 3.2 and 3.3, the $2dn + o(n)$ bound on the number of scalar comparisons holds when three conditions (a), (b), and (c) are satisfied. While (a) and (c) are unaffected by algorithm M itself, it takes more work to prove condition (b), which limits the running time of algorithm M on S_O to $o(n)$. From Lemma 3.8, $O(k|S_O|^{1+\epsilon}) = o(n)$ fails with probability at most $o(n^{-n^{\gamma}})$. Since M fails to achieve running time $O(k|S_O|^{1+\epsilon})$ with probability at most $1-p$, (b) fails with probability at most $\Theta\Big(\max\Big(1-p, o\Big(n^{-n^{\gamma}}\Big)\Big)\Big)$. Combining with fail probabilities of (a) and (c) from previous sections, Theorem 3.11 produces a convex hull algorithm that uses $2dn + o(n)$ scalar comparisons with probability at least $\Theta\Big(\min\Big(p, 1 - o\Big(n^{-n^{\gamma}}\Big)\Big)\Big)$. Since we prove later that M achieves a running time of $O(kn \log^{d-2} n)$ with probability at least $1 - \frac{1}{n^{1/d-\epsilon}}$, Theorem 3.13 follows by setting $c = 1$ and $\epsilon = 1/(2d) - 1/(d^2 + 2)$ as in Theorem 3.11.

What left is to analyze the running time of algorithm M. Let Q be the convex hull while $Q' = \bigcup_{\sigma \in \{+1,-1\}^d} L^{\sigma}$ be the *orthant hull*, $OH(S)$, of S. Bentley et al. proved that $Q \subseteq Q'$. Similar to other multi-layer definitions, for $i \geq 1$, let the ith orthant layer of S be the orthant hull of the set left after removing the first $i-1$ orthant layers from S. Let $OH^{(k)}(S)$ be the set of the first k orthant layers of the set S, where $OH^{(1)}(S)$ is sometimes simplified as $OH(S)$. Let $Q_1, Q_2, \ldots, Q_k$ be the first k convex layers of S and $S_i = S \setminus \bigcup_{j=1}^{i-1} Q_j$, where $S_1 = S$, be the corresponding subset of S after $CH^{(i-1)}(S)$ is removed from S, for all $1 \leq i \leq k$. We define Q'_i as $OH(S_i)$, $i \in [1, k]$, which is the *actual* orthant hull algorithm M uses. Since $Q_i \subseteq Q'_i$ from [2], a point inside the region enclosed by $OH^{(i)}(S)$ cannot be on $\bigcup_{j=1}^{i} Q'_j$. In other words, $\bigcup_{i=1}^{k} Q'_i$ is a subset of the points on the first k orthant layers[1] of S.

Each iteration i has two steps: first computing Q'_i by applying Kung et al.'s algorithm [31] once for each sign vector σ, then finding Q_i by Chazelle's algorithm [13]. While the first step takes $O(2^d n \log^{d-2} n) = O(n \log^{d-2} n)$ time, the second step uses $O(|Q'_i|^{\lfloor d/2 \rfloor})$ time. Summing over all k iterations, we get the following upper bound on the time

$$O(kn \log^{d-2} n + \sum_{i=1}^{k} |Q'_i|^{\lfloor d/2 \rfloor}) = O(kn \log^{d-2} n + k|Q''|^{\lfloor d/2 \rfloor})$$

[1] Note that $\bigcup_{i=1}^{k} Q'_i$ is not a subset of the first k *σ-maximal layers* of S, as shown in the remark.

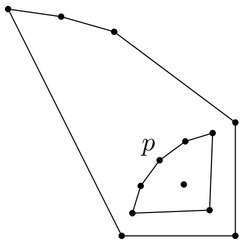


Figure 3.7: A counterexample showing the the first two convex layers is not a subset of a bounded number of σ-maximal layers.

where $Q'' = OH^{(k)}(S)$. As we show in Section 4.3,

$$E(k^{2/d}|Q''|) = O(k^{2/d}k^d \log^{d-1} n) = O(n^{1/d} \log^{d-1} n)$$

because $k \leq n^{1/(d^2+2)}$. Thus, by Markov's inequality, $P(k^{2/d}|Q''| > n^{2/d}) \leq n^{-1/d+\epsilon}$, that is, M takes $O(kn \log^{d-2} n)$ time with probability at least $1 - n^{-1/d+\epsilon}$, as claimed. □

Remark It is tempting to try to compute $k > 1$ convex layers by the first k σ-maximal layers of S (defined analogously to the σ-skyline of S). For this to work, one has to prove that the union of a bounded number of σ-maximal layers of S is a superset of the first k convex layers. However, this fails even for $k = 2$ because the second convex layer may contain a vertex that does not belong to the kth σ-maximal layer for any $\sigma \in \{+1, -1\}^d$ and any $k = o(n)$. Figure 3.7 shows an example in 2D where the point p is on the second convex layer but is in the $(\lfloor n/4 \rfloor + 1)$st σ-maximal layer for each $\sigma \in \{+1, -1\}^2$. This counterexample can be generalized to d dimensions to show that it is possible for a point in the second convex layer to be in the $(\lfloor n/2^d \rfloor + 1)$st σ-maximal layer for each $\sigma \in \{+1, -1\}^d$. It is possible that scenarios as in this figure are unlikely to arise in point sets drawn from a CI distribution, but this would require further analysis.

Chapter 4

Expected Size of the First k Layers

This chapter will focus on our second set of results, which is an upper bound on the expected size of the first k maximal or convex layers. The first maximal or convex layer of a point set drawn from a CI distribution has expected size $O(\log^{d-1} n)$, as proven in [2] by Bentley et al. Although it is hard to extend Bentley et al.'s technique to bound the size of any further layer, we are inspired by the counting approach in [2] to give in Section 4.1 a simple proof for an $O(\log n)$ bound on the expected size of a constant number of maximal layers in 2D.

For the result in general d dimensional space, we deeply thank an anonymous reviewer of a conference submission based on this work to provide a proof sketch bounding the size of maximal layers in 2D. We extend the sketch to show in Section 4.2 that, for continuous CI distributions in d dimensional space, the kth maximal layer has expected size $O((2k)^d \log^{d-1}(n/k^d))$, for any k in $[1, n]$. In Section 4.3, we show how to obtain the same bound (up to a factor of 8^d) for orthant layers, which is the key ingredient for Section 3.4. From the argument in Section 3.4, it is easy to deduct that $CH^{(k)}(S) \subseteq OH^{(k)}(S)$. Thus, Section 4.3 also gives $O(k^d \log^{d-1}(n/k^d))$ bound for the kth convex layer. Since it is also not hard to see that $SL^{(k)}(S) \subseteq OH^{(k)}(S)$, Section 4.2 and 4.3 could have been combined into one. However, we leave them as is to help the readers understand the idea more easily as the analysis on maximal layers is simpler. Because components of input points are drawn from continuous distribution, with probability 1, all points in the input have distinct coordinates in each dimension.

4.1 Constant Maximal Layers in 2D

This section proves that the expected size of the kth maximal layer in 2D is $O(\log n)$. Since Bentley et al. [2] already proved the result for $k = 1$, we focus on a constant integer $k \geq 2$. Let $S = \{p_1, p_2, \ldots, p_n\}$ and x_i, y_i denote the x- and y-coordinates of p_i for $1 \leq i \leq n$. Without loss of generality, we assume that S is sorted by x-coordinates such that $x_1 > x_2 > \cdots > x_n$. Because S is randomly picked from a CI distribution, the sequence $L_n = \langle y_1, y_2, \ldots, y_n \rangle$ is a uniform random permutation. Let K_i is the event that p_i belongs to the kth layer, $\forall i \in [1, n]$. Then, the expected size of the kth layer is $\sum_{i=1}^{n} Pr(K_i)$. By Lemma 4.1 below, $Pr(K_i) \leq \frac{c}{i}$, for some positive constant c. Thus, $\sum_{i=1}^{n} Pr(K_i) \leq \sum_{i=1}^{n} \frac{c}{i} =\leq c \sum_{i=1}^{n} \frac{1}{i} = O(\log n)$, which completes this section.

Lemma 4.1. *There exists a positive constant c such that $Pr(K_i) \leq \frac{c}{i}$.*

Proof. Since the depth of a point in maximal layers only depends on points dominating it, we are motivated to compute $Pr(K_i)$ through D_{ij}, which is the event that there are exactly j points that dominate p_i. Precisely,

$$Pr(K_i) = \sum_{j=1}^{n-1} Pr(K_i|D_{ij})Pr(D_{ij}) = \sum_{j=1}^{i-1} Pr(K_i|D_{ij})Pr(D_{ij}).$$

This identity holds because $Pr(D_{ij}) = 0$ for $j \geq i$, as only $i-1$ points to the right of p_i, which are $p_1, p_2, \dots, p_{i-1}$, can dominate p_i. Lemma 4.2 shows that $Pr(D_{ij}) = \frac{1}{i}$ for all $1 \leq j < i$. Lemma 4.3 shows that $Pr(K_i|D_{ij}) \leq \frac{4^{j(k-2)}}{j!}$. Thus,

$$Pr(K_i) \leq \frac{1}{i} \sum_{j=1}^{i-1} \frac{4^{j(k-2)}}{j!} \leq \frac{1}{i} e^{4^{k-2}}$$

because

$$\sum_{j=0}^{\infty} \frac{x^j}{j!} = e^x$$

Since k is a constant, $c = e^{4^{k-2}}$ is a positive constant. Thus, $Pr(K_i) \leq \frac{c}{i}$. □

What left is to prove Lemma 4.2 and 4.3.

Lemma 4.2. $Pr(D_{ij}) = \frac{1}{i}$.

Proof. Because S is x-sorted, only the element in $S_i = \{p_1, p_2, \dots, p_{i-1}, p_i\}$ can dominate p_i. S_i corresponds to the subsequence $L_i = \langle y_1, y_2, \dots, y_i \rangle$ of L_n. A point in S_i dominates p_i if and only if its y-coordinate is greater than y_i. Thus, D_{ij} is equivalent to the event that y_i is the $(j+1)$st largest element in L_i. Since L_n is a uniform random permutation, the subsequence L_i must also be a uniform random permutation. We have $Pr(D_{ij}) = \frac{1}{i}$. □

Lemma 4.3. $Pr(K_i|D_{ij}) \leq \frac{4^{j(k-2)}}{j!}$.

Proof. Let D be the subset of j points in $S_{i-1} = \{p_1, p_2, \ldots, p_{i-1}\}$ that dominate p_i. We observe that p_i belongs to the kth layer if and only if the point set D has $k-1$ maximal layers. We also observe that D is a CI random point set. Let $p_{m,t}$ be the probability that a CI random point set D' of size m has t layers. W.l.o.g, we assume that D' is sorted by x-coordinate in decreasing order. Let $g_{m,t}$ be the number of permutations of the y-coordinates where D' has t layers. We have $p_{m,t} = \frac{g_{m,t}}{m!}$. We will prove by induction on t that $g_{m,t} \leq 4^{m(t-1)}$. Thus, $p_{m,t} = \frac{g_{m,t}}{m!} \leq \frac{4^{m(t-1)}}{m!}$. It follows that $Pr(K_i|D_{ij}) = p_{j,k-1} \leq \frac{4^{j(k-2)}}{j!}$.

The base case is when $t = 1$. Since only the permutation arranging the y-coordinates in increasing order results in a single layer, $g_{m,t} = 1 = 4^{m(t-1)}$. Thus, the base case holds.

Suppose that $g_{m,t} \leq 4^{m(t-1)}$ for all values from 1 to $t-1$. We will show that the inequality also holds for t. Indeed, the event that D' has t maximal layers is equivalent to the event that D' can be partitioned into two subsets $D'_1 = SL(D')$ and $D'_2 = D' \setminus D'_1$. D'_2 has $t-1$ maximal layers. Since $SL(D')$ has one maximal layer in itself, $g_{m,t}$ is at most $n_{(1,t-1)}$, the number of permutations of the y-coordinates of the points in D' such that D' can be split into two subsets D'_1 with one layer and D'_2 with $t-1$ layers, regardless of whether or not D'_1 is the skyline of D'. To count $n_{(1,t-1)}$, we first fix the size of D'_1 to some value s. We pick s points for D'_1 and then pick s y-coordinates for these points. The remaining points and the remaining y-coordinates go to D'_2. Since $g_{s,1}$ is the number of permutations of the y-coordinates where D'_1 has 1 layer and $g_{m-s,t-1}$ is the number of permutations of the y-coordinates where D'_2 has $t-1$ layers, we have:

$$g_{m,t} \leq n_{(1,t-1)} = \sum_{s=1}^{m-t+1} \binom{m}{s}\binom{m}{s} g_{s,1} g_{m-s,t-1} = \sum_{s=1}^{m-t+1} \binom{m}{s}^2 g_{m-s,t-1}.$$

because $g_{s,1} = 1$ from the base case and D'_1 can have any size between 1 and $m-t+1$.

By the inductive hypothesis, $g_{m-s,t-1} \leq 4^{(m-s)(t-2)} \leq 4^{m(t-2)}$, this gives:

$$g_{m,t} \leq \sum_{s=1}^{m-t+1} \binom{m}{s}^2 4^{m(t-2)} = 4^{m(t-2)} \sum_{s=1}^{m-t+1} \binom{m}{s}^2 \leq 4^{m(t-2)} \cdot 4^m = 4^{m(t-1)}.$$

where $\sum_{s=1}^{m-t+1} \binom{m}{s}^2 \leq (\sum_{s=0}^{m} \binom{m}{s})^2 = (2^m)^2 = 4^m$. □

4.2 The First k Maximal Layers

We now present an approach based on integrals to bound the expected size of the first k layers. Our expected bound on the first k maximal layers is stated in the following theorem:

Theorem 4.4. *For any point set S drawn from a continuous CI distribution in d dimensions, the kth maximal layer has expected size $O(k^d \log^{d-1}(n/k^d))$.*

Proof. For an arbitrary continuous CI distribution $\mathcal{D}$, each ith dimension has a cumulative distribution function F_i, for all $1 \leq i \leq d$. For any point set S drawn from $\mathcal{D}$, the mapping ϕ maps each point p in S to a point $\phi(p) = (F_1(x_1(p)), F_2(x_2(p)), \ldots, F_d(x_d(p)))$ in set S'. By definition, S' is drawn uniformly at random from the unit hypercube. Any cumulative distribution function is non-decreasing. However, since the CI distribution is continuous, F_i must be an increasing function for $\forall i \in [1, d]$. Therefore, ϕ is a one-to-one mapping and $p \in S$ σ-dominates $q \in S$ if and only if $\phi(p)$ σ-dominates $\phi(q)$. It follows that the upper bound on the expected size of maximal layers in S' also holds for S.

We now consider this problem and redefine S to be the set of n points drawn from a unit hypercube. Given set S, the *minimal* layer of S are the points in S that do not dominate any other points. The kth minimal layer of S is the minimal layer of the set left after removing the first $k-1$ minimal layers from S. For convenience, we will analyze minimal layers instead of maximal ones because the kth *minimal* layer of S is equivalent to the kth maximal layer via the transformation $(x_1, x_2, \ldots, x_d) \mapsto (1 - x_1, 1 - x_2, \ldots, 1 - x_d)$.

Let $ML^{(k)}(S)$ be the set of points in S and on the first k minimal layers. For every point $p \in \mathbb{R}^d$, let D_p be the part of the unit hypercube dominated by p, and let $|D_p| = \prod_{i=1}^d x_i(p)$ be the volume of D_p. For $t \in \mathbb{N}$, let B_t be the surface containing all points $p \in [0,1]^d$ such that $|D_p| = \prod_{i=1}^d x_i(p) = \frac{(2k)^d t}{n}$; see Figure 4.1. B_t divides the unit hypercube into upper region L_t^+, including $(1, \ldots, 1)$, and lower region L_t^-, including $(0, \ldots, 0)$. For $t \geq 0$, we want to bound the volume of $L_t = L_t^+ \cap L_{t+1}^-$, the region between B_t and B_{t+1}. Refer to Figure 4.2, L_t^- composes of the region below the surface B_t and a union of $d-1$ small hyperrectangles, $\frac{(2k)^d t}{n}$ volume each. Thus, the volume of L_t^- is bounded by

$$|L_t^-| = \int \cdots \int_R \frac{(2k)^d t}{n x_1 x_2 \cdots x_{d-1}} \mathbf{d}x_1 \ldots \mathbf{d}x_{d-1} + \frac{(d-1)(2k)^d t}{n},$$

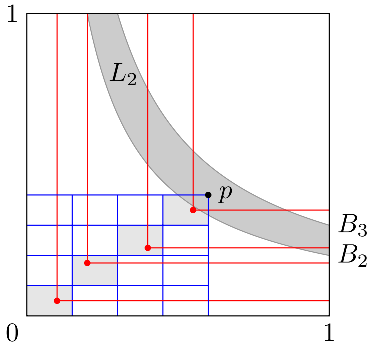


Figure 4.1: Bound on the probability that a point in L_t belongs to the first k minimal layers in 2D. Here, $k = 2$ and $n = 160$. The region L_2 is shaded. Each of the grey grid cells contains two points of S in expectation. Any point in such a grid cell (red) is dominated by p and all points in grid cells to its top right. Thus, unless more than two of these grid cells are empty, p does not belong to the first two minimal layers.

where $R = [\frac{(2k)^d t}{n}, 1] \times \ldots \times [\frac{(2k)^d t}{n}, 1]$ is a hypercube in $d - 1$ dimensions.

Since $f(x_1, \ldots, x_{d-1}) = \frac{(2k)^d t}{n x_1 x_2 \cdots x_{d-1}}$ is continuous on R, by Fubini's Theorem, the above multiple integral is indeed an iterated integral, which means that:

$$
\begin{aligned}
|L_t^-| &= \int_{(2k)^d t/n}^{1} \cdots \int_{(2k)^d t/n}^{1} \frac{(2k)^d t}{n x_1 x_2 \cdots x_{d-1}} \mathbf{d}x_1 \mathbf{d}x_2 \cdots \mathbf{d}x_{d-1} + \frac{(d-1)(2k)^d t}{n} \\
&= \int_{(2k)^d t/n}^{1} \cdots \int_{(2k)^d t/n}^{1} \left(\frac{(2k)^d t}{n x_2 \cdots x_{d-1}} \int_{(2k)^d t/n}^{1} \frac{1}{x_1} \mathbf{d}x_1 \right) \mathbf{d}x_2 \cdots \mathbf{d}x_{d-1} + \frac{(d-1)(2k)^d t}{n} \\
&= \int_{(2k)^d t/n}^{1} \cdots \int_{(2k)^d t/n}^{1} \left(\frac{(2k)^d t}{n x_2 \cdots x_{d-1}} \ln(x_1) \Big|_{(2k)^d t/n}^{1} \right) \mathbf{d}x_2 \cdots \mathbf{d}x_{d-1} + \frac{(d-1)(2k)^d t}{n} \\
&= \int_{(2k)^d t/n}^{1} \cdots \int_{(2k)^d t/n}^{1} \frac{(2k)^d t \ln(n/(2^d k^d t))}{n x_2 \cdots x_{d-1}} \mathbf{d}x_2 \cdots \mathbf{d}x_{d-1} + \frac{(d-1)(2k)^d t}{n} \\
&= O\left(\frac{(2k)^d t \log^{d-1}(n/k^d)}{n} \right).
\end{aligned}
$$

Because $L_t \subseteq L_{t+1}^-$, $|L_t| = O\Big(\frac{(2k)^d t \log^{d-1}(n/k^d)}{n}\Big)$. Now, we find the probability that a point $p \in L_t \cap S$ stays on one of the first k layers. Consider the hyperrectangle D_p and a uniform grid of it with $(2k)^d$ cells by dividing each side of D_p into $2k$ equal intervals. Because a point $p \in L_t$ will stay above the surface B_t, $|D_p| > \frac{(2k)^d t}{n}$. Thus, the volume of each cell in the grid is greater than t/n. It implies that in expectation, there are more than t points in each cell. Applying the Chernoff bound, any of these grid cells is empty with probability less than $e^{-t/4}$. The expected number of empty cell on the diagonal of the grid is at most

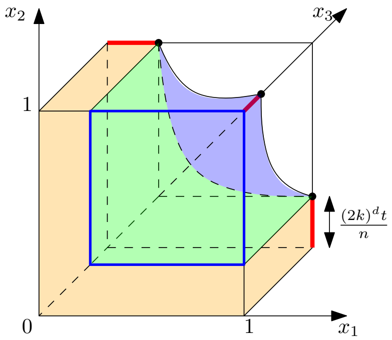


Figure 4.2: Illustration in 3D of L_t^-, which is the union of both the orange and green regions. The surface B_t is shaded blue. Each of the three red segments has length $\frac{(2k)^d t}{n}$. The blue square highlights the region over which the integral is computed.

$2ke^{-t/4}$. For p to be on one of the first k minimal layers, there must be at least k out of the $2k$ cells on the diagonal that are empty, which happens with probability less than $2e^{-t/4}$, by Markov's inequality. In other words, $Pr(p \in ML^{(k)}(S)|p \in L_t) < 2e^{-t/4}$. Then,

$$\begin{aligned} Pr(p \in ML^{(k)}(S) \wedge p \in L_t) &= Pr(p \in ML^{(k)}(S)|p \in L_t)Pr(p \in L_t) \\ &= O\left(\frac{(2k)^d t \log^{d-1}(n/k^d)}{ne^{t/4}}\right) \end{aligned}$$

where $Pr(p \in L_t) = |L_t| = O\Big(\frac{(2k)^d t \log^{d-1}(n/k^d)}{n}\Big)$, $k \leq n$ is a fixed parameter. The expected number of points in S that belong to L_t and to one of the first k layers is thus $O(\frac{(2k)^d t \log^{d-1}(n/k^d)}{e^{t/4}})$. Since the union of all L_t, $t \geq 0$, is the unit hypercube, summing over all L_t gives the expected number of points on the first k minimal layers, which is:

$$\begin{aligned} \sum_{t=0}^{\infty} O\left(\frac{(2k)^d t \log^{d-1}(n/k^d)}{e^{t/4}}\right) &= O((2k)^d \log^{d-1}(n/k^d)) \sum_{t=0}^{\infty} \frac{t}{e^{t/4}} \\ &= O((2k)^d \log^{d-1}(n/k^d)), \end{aligned}$$

since $\sum_{t=0}^{\infty} \frac{t}{e^{t/4}} \leq c$, for some positive constant c. Thus, the total expected size of the first k minimal layers, and therefore the expected size of the kth minimal layer, of S is $O((2k)^d \log^{d-1}(n/k^d))$. □

4.3 The First *k* Orthant Layers

In this section, we will bound the expected size of the first k orthant layers.

Theorem 4.5. *For any point set S drawn from a continuous CI distribution in d dimensions, the kth orthant layer has expected size $O(k^d \log^{d-1}(n/k^d))$.*

Proof. Similar to Section 4.2, we only need to prove the expected bound for inputs drawn from uniform distribution. Let o be the center point of the unit hypercube. Point o splits the unit hypercube into 2^d orthants O^σ with side length $1/2$ for $\sigma \in \{+1, -1\}^d$. Precisely, O^σ is the set of all points in the unit hypercube that σ-dominate o. As will be shown later, for every $\sigma \in \{-1, +1\}^d$, the expected size of $OH^{(k)}(S) \cap O^\sigma$ is $O((4k)^d \log^{d-1}(n/k^d))$. Summing over all 2^d orthants, we obtain a bound of $O((8k)^d \log^{d-1}(n/k^d)) = O(k^d \log^{d-1}(n/k^d))$ on the expected size of the first k orthant layers of S. Since there exists a one-to-one function mapping a point in an arbitrary orthant O^σ to another point in orthant $O^{-\mathbf{1}}$, w.l.o.g. we only need to analyze the orthant $O^{-\mathbf{1}}$. The argument for other orthant O^σ is symmetric.

We now start the inspection of orthant $O^{-\mathbf{1}}$ by defining some terminologies at first. For every point $p \in \mathbb{R}^d$ and for any $\sigma \in \{+1, -1\}^d$, let D_p^σ be the part of the unit hypercube σ-dominated by p. We observe that $D_p^{\mathbf{1}}$ is the D_p defined in Section 4.2. For $t \in \mathbb{N}$, let B_t be the surface containing all points $p \in [0, 1]^d$ such that $|D_p^{\mathbf{1}}| = \prod_{i=1}^d x_i(p) = \frac{(2k)^d t}{n}$. B_t divides the unit hypercube into upper region L_t^+, including $(1, \dots, 1)$, and lower region L_t^-, including $(0, \dots, 0)$. For $t \geq 0$, let $L_t = L_t^+ \cap L_{t+1}^-$ be the region between B_t and B_{t+1}.

We want to bound the probability that a point $p \in L_t \cap O^{-\mathbf{1}}$ belongs to one of the first k orthant layers. For a point $p \in L_t \cap O^{-\mathbf{1}}$ and for any $\sigma \in \{+1, -1\}^d$, we divide each hyperrectangle D_p^σ into a uniform grid with $(2k)^d$ cells by dividing each side of D_p^σ into $2k$ equal intervals. For all σ, starting from p to the opposite corner of D_p^σ, let $H_1^\sigma, H_2^\sigma, \dots, H_{2k}^\sigma$ be the grid cells on the diagonal in the same order. The argument in Section 4.2 shows that $Pr(S \cap H_i^{-\mathbf{1}} = \emptyset) < e^{-t/4}$ for all $1 \leq i \leq 2k$. Because $p \in O^{-\mathbf{1}}$, we have $|H_i^\sigma| \geq |H_i^{-\mathbf{1}}|$ for all i and all σ. Thus, the probability for any diagonal cell to be empty is $Pr(S \cap H_i^\sigma = \emptyset) \leq Pr(S \cap H_i^{-\mathbf{1}}) < e^{-t/4}$. For any fixed index $1 \leq i \leq 2k$, among 2^d all possible sign vectors, the probability that there exists at least one σ_i such that $S \cap H_i^{\sigma_i} = \emptyset$ is less than $2^d e^{-t/4}$. Therefore, $E[X] < 2^{d+1} k e^{-t/4}$, where X is the number of indices i such that $S \cap H_i^{\sigma_i} = \emptyset$ for some sign vector σ_i. By Markov's inequality, $Pr(X \geq k) < 2^{d+1} e^{-t/4}$. Similar to the proof of Lemma 3.9, if there exist k indices $1 \leq i_1 < i_2 < \dots < i_k \leq 2k$ such that, for all $1 \leq j \leq k$ and all $\sigma \in \{-1, +1\}^d$, $H_{i_j}^\sigma \cap S \neq \emptyset$, then p is not on the kth orthant layer. Thus, for p to be on the kth orthant layer, there have to be at least k indices $1 \leq i'_1 < i'_2 < \dots < i'_k \leq 2k$

and sign vectors $\sigma_1, \sigma_2, \ldots, \sigma_k$ such that $S \cap H_{i'_j}^{\sigma_j} = \emptyset$ for all $1 \leq j \leq k$. Therefore,

$$Pr(p \in OH^{(k)}(S) | p \in L_t \cap O^{-\mathbf{1}}) \leq Pr(X \geq k) < 2^{d+1} e^{-t/4}.$$

Since $L_t \cap O^{-\mathbf{1}} \subseteq L_t$, from the exact same calculation as in Section 4.2, $|L_t \cap O^{-\mathbf{1}}| \leq |L_t| = O\Big(\frac{(2k)^d t \log^{d-1}(n/k^d)}{n}\Big)$. We have

$$\begin{aligned} Pr(p \in OH^{(k)}(S) \wedge p \in L_t \cap O^{-\mathbf{1}}) &= Pr(p \in OH^{(k)}(S) | p \in L_t \cap O^{-\mathbf{1}}) Pr(p \in L_t \cap O^{-\mathbf{1}}) \\ &= O\Big(\frac{(4k)^d t \log^{d-1}(n/k^d)}{n e^{t/4}}\Big) \end{aligned}$$

where $k \leq n$ is a fixed parameter. Thus, the expected number of points in $L_t \cap O^{-\mathbf{1}}$ that belong to the first k orthant layers is $O\Big(\frac{(4k)^d t \log^{d-1}(n/k^d)}{e^{t/4}}\Big)$. Again, since $\sum_{t=0}^{\infty} \frac{t}{e^{t/4}} \leq c$ where c is a positive constant, summing over all L_t, $t \geq 0$, gives the expected number of points in $O^{-\mathbf{1}}$ that belong to the first k orthant layers, which is $O((4k)^d \log^{d-1}(n/k^d))$, as claimed. □

Chapter 5

Conclusions

This thesis studies two well-known geometric structures in computational geometry: maximal points and convex hull. Extending the concepts to multiple maximal and convex layers is natural. There have been extensive works around the topic. For worst-case analysis, the first maximal or convex layer can be computed optimally in [31], [4], [13], while all layers in two and three dimensions can be extracted by [5], [7], [12]. The result in [5] even achieves an in-place solution, which is optimal in space. For output sensitive analysis, multiple layers can be computed in 2D by [35]. A short adversary argument shows that at least $\Omega(n \log n)$ comparisons are required to compute even the first layer. However, for certain distribution, like the CI distribution, it is possible to use $O(n)$ expected comparisons. Bentley et al. [3] are the first to present algorithms to compute the first maximal and convex layer from CI distribution in $dn + o(n)$ and $2dn + o(n)$ expected scalar comparisons, respectively. The result in [3] is also interesting in the aspect of optimizing the leading term of the number of comparisons, which is an attractive field of its own with many papers, such as [42], [11] and [16].

We are the first to present, in the first part of this thesis, algorithms to compute multiple maximal and convex layers using near optimal expected number of scalar comparisons. The final result is even lifted to high probability. The algorithms find a certificate to prune most of the points. The certificate is chosen such that the remaining set is likely to contain the first k layers. Therefore, it is cheap to extract multiple layers using known worst-case algorithms. Although it is not hard to extend the algorithms in [3] to compute inner layers, it is challenging to analyze the running time. Our analysis is based on an idea of diagonal-grid building, which proves to be useful for other purposes; for example, we apply the same idea to bound the size of the first k layers in the second half of the thesis.

The rest of the thesis studies the complexity of maximal and convex layers. Analyzing complexities of geometric structures is interested by many researchers, as shown in [39], [40], etc. The expected size of the first layer over an arbitrary CI distribution is proven to be

$O(\log^{d-1} n)$ by [2]. Although we cannot extend the technique in [2] to bound any further layer, we are inspired by the counting method to give a bound for a constant number of maximal layers in 2D. With the help from an anonymous reviewer, we show that the first k layers have expected size $O(k^d \log^{d-1}(n/k^d))$. To our surprise, the upper bound in this part enables us to compute multiple convex layers in high dimensions when $d \geq 4$. So far, there is no expected linear-time algorithm to compute even the second convex layers when $d \geq 4$ because of the high convex-hull complexity. We are the first to compute a non-constant number of convex layers in high probability using at most twice the optimal number of scalar comparisons.

Several open problems arise from the thesis.

Open Problem 1. *Either tighten the bound $O(k^d \log^{d-1}(n/k^d))$ on the expected size of the first k maximal and convex layers or prove a matching lower bound.*

Open Problem 2. *The constant factor in the bound on the expected size of the first k convex layers currently is 4^d. From the nature of the relationship between maximal and convex layers shown in this thesis, one would expect the factor to be 2^d. Is it possible to achieve that?*

Bibliography

[1] Peyman Afshani, Jérémy Barbay, and Timothy M. Chan. Instance-optimal geometric algorithms. *J. ACM*, 64(1):3:1–3:38, March 2017.

[2] J. L. Bentley, H. T. Kung, M. Schkolnick, and C. D. Thompson. On the average number of maxima in a set of vectors and applications. *J. ACM*, 25(4):536–543, 1978.

[3] Jon L. Bentley, Kenneth L. Clarkson, and David B. Levine. Fast linear expected-time algorithms for computing maxima and convex hulls. *Algorithmica*, 9(2):168–183, 1993.

[4] Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 3rd edition, 2008.

[5] Henrik Blunck and Jan Vahrenhold. In-place algorithms for computing (layers of) maxima. In Lars Arge and Rusins Freivalds, editors, *Proc. 10th Scand. Workshop on Algorithm Theory*, volume 4059 of *LNCS*, pages 363–374. Springer-Verlag, 2006.

[6] Stephan Börzsönyi, Donald Kossmann, and Konrad Stocker. The skyline operator. In *Proc. 17th International Conference on Data Engineering*, pages 421–430, 2001.

[7] Adam L. Buchsbaum and Michael T. Goodrich. Three-dimensional layers of maxima. *Algorithmica*, 39(4):275–286, 2004.

[8] Andrew Butterfield, Gerard Ekembe Ngondi, and Anne Kerr. *A Dictionary of Computer Science (7 ed.)*. Oxford University Press, 7th edition, 2016.

[9] T. M. Chan. Optimal output-sensitive convex hull algorithms in two and three dimensions. *Discrete Comput. Geom.*, 16(4):361–368, April 1996.

[10] T. M. Chan, J. Snoeyink, and Chee-Keng Yap. Primal dividing and dual pruning: Output-sensitive construction of four-dimensional polytopes and three-dimensional voronoi diagrams. *Discrete & Computational Geometry*, 18(4):433–454, Dec 1997.

[11] Timothy M. Chan and Patrick Lee. On constant factors in comparison-based geometric algorithms and data structures. *Discrete Comput. Geom.*, 53(3):489–513, April 2015.

[12] B. Chazelle. On the convex layers of a planar set. *IEEE Trans. Inf. Theor.*, 31(4): 509–517, 2006.

[13] Bernard Chazelle. An optimal convex hull algorithm in any fixed dimension. *Discrete Comput. Geom.*, 10(4):377–409, 1993.

[14] Bernard Chazelle and J. Matoušek. Derandomizing an output-sensitive convex hull algorithm in three dimensions. *Computational Geometry*, 5(1):27 – 32, 1995.

[15] Bernard Chazelle, Leo J. Guibas, and D. T. Lee. The power of geometric duality. *BIT*, 25(1):76–90, June 1985.

[16] K. L. Clarkson. More output-sensitive geometric algorithms. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, SFCS '94, pages 695–702, Washington, DC, USA, 1994. IEEE Computer Society. ISBN 0-8186-6580-7.

[17] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, ii. *Discrete Comput. Geom.*, 4(5):387–421, September 1989. ISSN 0179-5376.

[18] H. K. Dai and X. W. Zhang. *Improved Linear Expected-Time Algorithms for Computing Maxima*, pages 181–192. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004.

[19] Ketan Dalal. Counting the onion. *Random Struct. Algorithms*, 24(2):155–165, 2004. ISSN 1042-9832.

[20] Dorit Dor and Uri Zwick. Selecting the median. In *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '95, pages 28–37, Philadelphia, PA, USA, 1995. Society for Industrial and Applied Mathematics.

[21] Dorit Dor and Uri Zwick. Median selection requires (2+epsilon)n comparisons. *SIAM Journal on Discrete Mathematics*, 14(3):312–325, 2001.

[22] Lester R. Ford and Selmer M. Johnson. A tournament problem. *The American Mathematical Monthly*, 66(5):387–389, 1959.

[23] Alan Frieze. On the length of the longest monotone subsequence in a random permutation. *The Annals of Applied Probability*, 1(2):301–305, 1991.

[24] M. Golin and R. Sedgewick. Analysis of a simple yet efficient convex hull algorithm. In *Proceedings of the Fourth Annual Symposium on Computational Geometry*, SCG '88, pages 153–163, New York, NY, USA, 1988. ACM.

[25] M. J. Golin. A provably fast linear-expected-time maxima-finding algorithm. *Algorithmica*, 11(6):501–524, 1994.

[26] Gaston H. Gonnet and J. Ian Munro. *Heaps on heaps*, pages 282–291. Springer Berlin Heidelberg, Berlin, Heidelberg, 1982.

[27] Sariel Har-peled. On the expected complexity of random convex hulls. Technical report, 1997.

[28] David G. Kirkpatrick and Raimund Seidel. Output-size sensitive algorithms for finding maximal vectors. In *Proceedings of the First Annual Symposium on Computational Geometry*, SCG '85, pages 89–96, New York, NY, USA, 1985. ACM.

[29] David G Kirkpatrick and Raimund Seidel. The ultimate planar convex hull algorithm. *SIAM J. Comput.*, 15(1):287–299, February 1986.

[30] Donald E. Knuth. *The Art of Computer Programming, Volume 3: (2Nd Ed.) Sorting and Searching.* Addison Wesley Longman Publishing Co., Inc., Redwood City, CA, USA, 1998.

[31] H. T. Kung, F. Luccio, and F. P. Preparata. On finding the maxima of a set of vectors. *J. ACM*, 22(4):469–476, 1975.

[32] Zhentao Li and Bruce A. Reed. *Heap Building Bounds*, pages 14–23. Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.

[33] J. Matoušek and P. Plecháč. On functional separately convex hulls. *Discrete & Computational Geometry*, 19(1):105–130, Jan 1998. ISSN 1432-0444.

[34] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, New York, NY, USA, 1995. ISBN 0-521-47465-5, 9780521474658.

[35] Franck Nielsen. Output-sensitive peeling of convex and maximal layers. *Information Processing Letters*, 59(5):255–259, 1996.

[36] Atsuyuki Okabe, Barry Boots, Kokichi Sugihara, Sung Nok Chiu, and D. G. Kendall. *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams, 2nd Edition.* John Wiley & Sons, Inc., 2008. ISBN 9780470317013.

[37] P. K. Agarwal. Personal communication with Buchsbaum and Goodrich, 1992.

[38] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction.* Springer-Verlag New York, Inc., New York, NY, USA, 1985.

[39] H. Raynaud. Sur l'enveloppe convexe des nuages de points aleatoires dans rn. i. *Journal of Applied Probability*, 7(1):3548, 1970.

[40] A. Rényi and R. Sulanke. über die konvexe hülle von n zufällig gewählten punkten. *Zeitschrift für Wahrscheinlichkeitstheorie und Verwandte Gebiete*, 2(1):75–84, Jan 1963.

[41] Raimund Seidel. A convex hull algorithm optimal for point sets in even dimensions. Technical report, Vancouver, BC, Canada, Canada, 1981.

[42] Raimund Seidel and Udo Adamy. On the exact worst case query complexity of planar point location. *J. Algorithms*, 37(1):189–217, October 2000.

[43] Wolfgang Weil and John A. Wieacker. *Stochastic Geometry*, page 13931438. North Holland, 1993.

[44] R. Wenger. Randomized quickhull. *Algorithmica*, 17(3):322–329, Mar 1997.

Appendix A

LazySelect for Elements Far From the Median

Our algorithm for CI distributions makes use of the following theorem, which we prove here.

Theorem A.1. *For any $t \in (0,1)$ and any $k < n^{1-t}$, the kth largest element in a set S of n elements can be found using $n + o(n)$ comparisons with probability $1 - o(n^{-n^{\epsilon}})$, for any constant $\epsilon \in (0,1)$ and sufficiently large n. The expected number of comparisons is also $n + o(n)$. Furthermore, every element of S can be tagged as an element greater than, equal to, or less than the kth largest element of S without incurring additional comparisons.*

We simplify the LazySelect algorithm in [34] to prove Theorem A.1. Let $m = o(n/\log n)$ and $0 < \alpha < 1$ be parameters to be chosen later. We sample a set R of m elements from S with replacement. Since m is small, we can sort R by any optimal worst-case sorting algorithm using $o(n)$ comparisons. Then, instead of picking two *pivots* as in [34], we only set one, $a = R[\ell]$, where $\ell = (1-\alpha)m$. By comparing every element in S to a and partition S into two sets $S_<$ and $S_\geq$ containing the elements less than and greater than or equal to a, respectively. This takes n comparisons. As we shall prove later, it is likely that the kth largest element will fall in $S_\geq$ while $|S_\geq|$ is relatively small. Precisely, we sort $S_\geq$ using $o(n)$ comparisons and select the $(k - |S_<|)$th element x in $S_\geq$ if $|S_<| < n - k$ and $|S_\geq| \leq \frac{n}{\log^2 n}$. Otherwise, we repeat this process until $|S_<| < n - k$ and $|S_\geq| \leq \frac{n}{\log^2 n}$. The tagging of elements in S as being less than, or greater than or equal to x, the kth largest element of S, required in Section 3.2 is easily accomplished without additional comparisons by tagging all elements in $S_<$ and preceding x in $S_\geq$ as less than x and the remaining elements in $S_\geq$ as greater than or equal to x.

By Lemma A.2 and A.3, the probability for either $|S_<| \geq n - k$ or $|S_\geq| > \frac{n}{\log^2 n}$ to happen is at most $o(n^{-n^{\epsilon}})$. Thus, with high probability at least $1 - o(n^{-n^{\epsilon}})$, the algorithm will stop after the first iteration, which means that it only uses $n + o(n)$ comparisons. The expected number of iterations the algorithm executes is $\sum_{i=0}^{\infty} \left(o(n^{-n^{\epsilon}})\right)^i = 1 + o(1)$. It follows that the expected number of comparisons the algorithm performs is $n + o(n)$. This proves Theorem A.1. What left is the proof of Lemma A.2 and A.3.

Lemma A.2. *For any* $\alpha > n^{-t}$ *and* $k < n^{1-t}$, $Pr(|S_<| \geq n - k) < \left(\frac{e}{\alpha n^t}\right)^{\alpha m}$.

Proof. Let x be the kth largest element in S and X be the number of elements in R greater than x. If $|S_<| \geq n - k$, then $x \in S_<$ and thus $x < a$. Since $a = R[(1-\alpha)m]$, $x < a$ is equivalent to $X > \alpha m$. Therefore,

$$Pr(|S_<| \geq n - k) \leq Pr(X > \alpha m)$$

Because the probability for an element in S to be greater than x is $(k-1)/n$ and $|R| = m$, we have:

$$E(X) = \frac{k-1}{n} m < \frac{k}{n} m < \frac{m}{n^t}$$

if $k < n^{1-t}$. By Chernoff bound,

$$Pr(X > (1+\delta)E(X)) < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^{E(X)} \leq \left(\frac{e}{1+\delta}\right)^{(1+\delta)E(X)}$$

We choose $\delta = \alpha n^t - 1 > 0$. Since $E(X) < \frac{m}{n^t}$,

$$Pr(|S_<| \geq n - k) \leq Pr(X > \alpha m) < \left(\frac{e}{\alpha n^t}\right)^{\alpha m}$$

□

Lemma A.3. *For any* $\alpha < \frac{1}{\log^2 n}$, $Pr(|S_\geq| > \frac{n}{\log^2 n}) < e^{\frac{-(1-\alpha \log^2 n)^2 m}{2 \log^2 n}}$.

Proof. Let y be the $(\frac{n}{\log^2 n})$th largest element in S and Y be the number of elements in R at least y. If $|S_\geq| > \frac{n}{\log^2 n}$, then $y \in S_\geq$ and $y > a$. Since $a = R[(1-\alpha)m]$, $y > a$ is equivalent to $Y \leq \alpha m$. Therefore,

$$Pr(|S_\geq| > \frac{n}{\log^2 n}) \leq Pr(Y \leq \alpha m)$$

Because the probability for an element in S to be greater than y is $(\frac{1}{\log^2 n})$ and $|R| = m$, we have:

$$E(Y) = \frac{m}{\log^2 n}$$

Apply Chernoff bound with $1 > \delta = 1 - \alpha \log^2 n > 0$ for $\alpha < \frac{1}{\log^2 n}$,

$$Pr(Y < (1-\delta)E(Y)) < e^{\frac{-\delta^2 E(Y)}{2}}$$

$$\Longrightarrow Pr(|S_\geq| \geq \frac{n}{\log^2 n}) \leq Pr(Y < \alpha m) < e^{\frac{-(1-\alpha \log^2 n)^2 m}{2 \log^2 n}}$$

□

Choosing $\alpha = \frac{1}{2\log^2 n}$ satisfies both conditions on α in Lemmas A.2 and A.3 for large enough n and gives $Pr(|S_<| \geq n-k) < \left(\frac{2e\log^2 n}{n^t}\right)^{\frac{m}{2\log^2 n}}$ and $Pr(|S_\geq| \geq \frac{n}{\log^2 n}) < e^{-m/8\log^2 n}$ because $\alpha = \frac{1}{2\log^2 n}$ implies that $(1-\alpha\log^2 n)^2 = 1/4$. Both bounds are minimized when m is maximized, so we choose $m = \frac{n}{\log^2 n}$. Since $t > 0$, $\frac{n^t}{2e\log^2 n} > e$ for sufficiently large n, we obtain that both $Pr(|S_<| \geq n-k) < e^{-n/2\log^4 n} = o(n^{-n^\epsilon})$ and $Pr(|S_\geq| \geq \frac{n}{\log^2 n}) < e^{-n/8\log^4 n} = o(n^{-n^\epsilon})$, for any $0 < \epsilon < 1$. This finishes the proof of Theorem A.1.