EINDHOVEN UNIVERSITY OF TECHNOLOGY

Separability of imprecise points

Citation for published version (APA):
Sheikhi, F., Mohades , A., de Berg, M. T., & Mehrabi Davoodabadi, A. (2017). Separability of imprecise points. *Computational Geometry*, *61*, 24–37. https://doi.org/10.1016/j.comgeo.2016.10.001

Document license:
TAVERNE

DOI:
10.1016/j.comgeo.2016.10.001

Document status and date:
Published: 01/02/2017

Document Version:
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Download date: 02. Jun. 2023

Computational Geometry 61 (2017) 24–37

Contents lists available at ScienceDirect

Computational Geometry: Theory and Applications

www.elsevier.com/locate/comgeo

Separability of imprecise points ☆

Farnaz Sheikhi [a], Ali Mohades [a,∗,1], Mark de Berg [b,1], Ali D. Mehrabi [b]

[a] Computer Science Department, Amirkabir University of Technology, Iran
[b] Department of Computer Science, Eindhoven University of Technology, the Netherlands

ARTICLE INFO

Article history:
Received 27 March 2015
Accepted 18 October 2016
Available online 16 November 2016

Keywords:
Imprecise points
Separator

ABSTRACT

An imprecise point p in the plane is a point represented by an imprecision region $\mathcal{I}_p$ indicating the set of possible locations of the point p. We study separability problems for a set R of red imprecise points and a set B of blue imprecise points, where the imprecision regions are axis-parallel rectangles and each point $p \in R \cup B$ is drawn uniformly at random from $\mathcal{I}_p$. Our results include algorithms for finding *certain separators* (which separate R from B with probability 1), *possible separators* (which separate R from B with non-zero probability), *most likely* separators (which separate R from B with maximum probability), and *maximal* separators (which maximize the expected number of correctly classified points).

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Separability problems are a natural class of problems arising in the analysis of categorical geometric data. In a bichromatic separability problem one is given a set of n points in $\mathbb{R}^d$, each of which is categorized as either *red* or *blue*, and the goal is to decide whether the red points can be separated from the blue points by a *separator* from a given class of geometric objects. If a separator exists, one may also be interested in finding all separators or the separator minimizing a given cost function. In this paper we are interested in separability problems on *imprecise points*, that is, points of which the location is not know precisely. In the remainder of this introduction we first give an overview of the known results on separability problems for normal (that is, precise) points, then we discuss some existing work from computational geometry on imprecise points, and finally we state our results on separability problems for imprecise points.

Separability problems. For separators in the form of a hyperplane, the decision version of the separability problem can be solved by linear programming in $O(n)$ time, when the dimension d is a fixed constant, as was observed by Megiddo [23] already 30 years ago. Since then various classes of separators have been studied, mostly for the 2-dimensional version of the problem. O'Rourke et al. [27] studied separability problems for circular separators. They proposed an $O(n)$-time algorithm for deciding whether such a separator exists. Furthermore, they showed that the smallest and the largest separating circle (if they exist) can be found optimally in $O(n)$ and $O(n \log n)$ time, respectively. The problem of finding a convex polygon with minimum number of edges separating the two point sets was solved by Edelsbrunner and Preparata [10] in $O(n \log n)$

☆ A preliminary version appeared in the 14th Scandinavian Symposium and Workshop on Algorithm Theory (SWAT) 2014.

∗ Corresponding author.
E-mail address: mohades@aut.ac.ir (A. Mohades).

1 M. de Berg and A. D. Mehrabi were supported by the Netherlands Organization for Scientific Research (NWO) under grants 024.002.003 and 612.001.118, respectively.

http://dx.doi.org/10.1016/j.comgeo.2016.10.001
0925-7721/© 2016 Elsevier B.V. All rights reserved.

time. Fekete [12] showed that separating two point sets by a simple polygon with minimum number of edges is NP-hard, and a polynomial-time $O(\log n)$-approximation algorithm was presented by Mitchell [24].

Other results on the separability problem concern strips and wedges as separators [16,2,29]. Hurtado et al. [16] showed that deciding whether two point sets can be separated by a strip or a wedge can be done in $O(n \log n)$ time, which is optimal [2]. Hurtado et al. [16] also showed that all orientations of the separating strips can be found in $O(n \log n)$ time; after this, the separating strip with the minimum and maximum width can be computed in $O(n)$ and $O(n \log n)$ time, respectively. Moreover, they showed that the separating wedge with the minimum and maximum angle can be computed in $O(n \log n)$ time. A thorough study of these types of separators is presented by Seara [29].

Inspired by the reconstruction of buildings from LIDAR[2] data, Van Kreveld et al. [17] presented an $O(n \log n)$-time algorithm to compute all orientations for which a separating rectangle exists. This was extended to L-shaped separators by Sheikhi et al. [30], who presented a worst-case optimal $O(n^2)$-time algorithm to compute all orientations for which an L-shaped separator exists. They also gave an output-sensitive algorithm running in $O(n^{8/5+\varepsilon} + k \log k)$ time, where k is the number of reported orientations and $\varepsilon > 0$ is any fixed constant.

Obviously it is not always possible to separate the given point sets by a separator of the given type. Houle [14,15] therefore introduced *weak separability*, where the goal is to maximize the number of correctly classified points. For example, for linear separability the weak separability problem asks for a line ℓ that maximizes the sum of the number of red points to the right of ℓ and the number of blue points to the left of ℓ. (A separator that correctly classifies all points is then called a *strong separator*.) For linear separators, Houle [14] showed that the weak separability problem can be solved in $O(n^2)$ time. Following that, Everett et al. [11] solved this problem in $O(nk \log k + n \log n)$ time, where k is the number of misclassified points. Chan [6] presented an $O((n + k^2) \log n)$-expected time algorithm for this problem, and Bereg et al. [4] showed that the problem is 3SUM-hard. Cortés et al. [7] presented an $O(n^2 \log n)$-time algorithm for the weak separability problem for a strip. Aronov et al. [3] studied the problem of measuring the quality of a weak separator according to how much work it would take (based on several distance functions) to move the misclassified points across the separator in order to make it a strong separator.

Imprecise points. In data-analysis problems involving geometric data, the data is typically obtained by GPS, LIDAR, or some other imprecise measuring technology. Ideally, one would like to take this imprecision into account when analyzing the data. Within the computational-geometry literature, several models have been proposed to handle imprecision [8,18,25]. The most popular models associate with each data point p an *imprecision region* $\mathcal{I}_p$, which indicates the set of possible locations of p. Typical choices for the imprecision regions are disks [28], axis-parallel rectangles or squares [18], and horizontal segments [18]. Horizontal segments model the situation where there is imprecision in only one of the coordinates, and rectangles or squares model the situation where the coordinates come from independent measurements. A point p with an associated imprecision region $\mathcal{I}_p$ is often called an *imprecise point*. Löffler [18] and Löffler and Van Kreveld [19,20] studied several classical computational-geometry problems on imprecise points. In most problems they wanted to find certain "extremal" structures, such as the largest or smallest possible convex hull. De Berg et al. [5], on the other hand, studied the question whether a given structure is possible (for example, whether a given subset of the points could be the convex hull).

Some recent work on imprecise (or: uncertain) points considers probabilistic aspects. Suri et al. [31] presented two models for this. In the *unipoint model* each point has a fixed location and an associated probability of existence. In other words, one is given a set of potential locations, and at each potential location a point exists with a certain probability. In the *multipoint model*, each point has several potential locations with an associated probability. Suri et al. then studied the problem of finding the *most likely convex hull*, that is the convex hull with the maximum probability of occurrence. In the unipoint model they proposed an $O(n^3)$-time algorithm for dimension $d = 2$, and showed that the problem is NP-hard for $d \geqslant 3$. In the multipoint model the problem is NP-hard even for $d = 2$. Agarwal et al. [1] proposed exact and approximation algorithms to compute the probability of a query point lying inside the convex hull of the input.

Our results. In this paper we study various separability problems for imprecise points. We extend the region-based imprecision model to include probabilistic aspects. More precisely, we assume each point p is drawn from its imprecision region $\mathcal{I}_p$ according to some distribution. In the current paper, we consider the uniform distribution. Throughout the paper we assume all the input imprecision regions are (relatively) open regions. Given a set R of red imprecise points and a set B of blue imprecise points, and a class of separators, we then wish to find

- a *certain separator*, which separates R from B with probability 1;
- a *possible separator*, which separates R from B with non-zero probability;
- a *most likely separator*, which separates R from B with maximum probability;
- a *maximal separator*, which is a weak separator that maximizes the expected number of correctly classified points.

In these problems, we require our separators to be strict, that is, we do not allow points on the separator. Thus (since the imprecision regions are open) a certain separator is a separator such that for any choice of points from the imprecision

[2] LIDAR stands for LIght Detection and Ranging, a popular remote-sensing technology.

Table 1
Overview of our results in the plane. The imprecision regions are axis-parallel rectangles, except for HS which stands for horizontal segments as imprecision regions. The results labeled "approx" provide a $(1-\varepsilon)$-approximation.

Separator	Certain	Possible	Maximal
Line	$O(n)$	$O(n\log n)$ decision	$O(n^2)$ exact
		$O(n\log n)$ find all	$O(\frac{n}{\varepsilon^6}\log^2\frac{1}{\varepsilon})$ approx
Axis-parallel	$O(n)$	$O(n)$ decision	$O(n^3\log n)$ exact
Rectangle exact (HS)		$O(n^2\log n)$ find all	$O(n^2\sqrt{n})$
			$O(\frac{n}{\varepsilon^9}\log^5\frac{1}{\varepsilon})$ approx

regions the separator is a strict separator. Similarly, a possible separator is a separator that is a strict separator for some choice of points.

Most of our results are for the case where the imprecision regions are axis-parallel rectangles. (We do not require the rectangles to be of the same size or aspect ratio.) Our results are as follows. In Section 2 we first observe that finding a certain separator can easily be done in $O(n)$ time, both for linear separators and for rectangular separators. Finding possible separators is fairly easy as well: a possible linear separator can be found in $O(n\log n)$ time, while a possible rectangular separator can be found in $O(n)$ time. We also show how to compute all possible linear separators in $O(n\log n)$ time. For rectangular separators we show how to compute all possible separators in $O(n^2\log n)$ time, which is close to optimal since there can be $\Omega(n^2)$ combinatorially distinct possible separators. Most likely separators are a lot harder. Here we study the 1-dimensional case, which already turns out to be hard to solve, since it requires finding the maximum of a possibly high-degree polynomial.

In Section 3 we then turn our attention to weak separability. We present exact algorithms for linear separators (running in $O(n^2)$ time), for rectangular separators (running in $O(n^3\log n)$ time), and for rectangular separators when the imprecision regions are horizontal segments (running in $O(n^2\sqrt{n})$ time). We also present $O(n\cdot\text{poly}(1/\varepsilon))$-time $(1-\varepsilon)$-approximation algorithms for weak separability for linear and rectangular separators; here $\text{poly}(1/\varepsilon)$ denotes a polynomial in $1/\varepsilon$. Our results are summarized in Table 1.

2. Strong separability

Let R be a set of red points and B be a set of blue points in the plane, with $n := |R|+|B|$. Each point $p\in R\cup B$ has an associated imprecision region $\mathcal{I}_p$, which is an axis-parallel rectangle. In this section we give algorithms to find strong separators, that is, separators that classify all points correctly.

2.1. Certain and possible separators

Certain separators. A line (or other shape) is a certain separator if and only if all the red imprecision regions lie entirely on one side of it while all the blue imprecision regions lie entirely on the other side. Hence, deciding whether $R\cup B$ admits a certain separator is very easy: a line ℓ is a certain separator if and only if the vertices of the red and blue imprecision regions lie in opposite (closed) half-planes defined by ℓ, and so we can decide the existence of a certain separator by linear programming. Finding a rectangular certain separator is also easy: if there is an axis-parallel rectangle with, say, all red imprecision regions inside and all blue imprecision regions outside, then the bounding box of the red imprecision regions is a certain separator. Hence, we can test this in linear time. It is worth mentioning that we can also find an orthoconvex certain separator in optimal $O(n\log n)$ time by using the results of Ottmann et al. [26] as follows. Compute the orthoconvex hull of the set of all red corners in optimal $O(n\log n)$ time using the results of [26]. Then use a plane-sweep algorithm to test in $O(n\log n)$ time if all the blue imprecision regions are outside of the orthoconvex hull.

Possible separators. Finding possible separators is only slightly more involved than finding certain separators. First, consider linear separators. We wish to find a possible separator ℓ (which we consider to be a directed line) that has the red points to its left and the blue points to its right. Then ℓ is a possible separator unless there is a red imprecision region lying completely to the right of ℓ or a blue imprecision region lying completely to the left. Thus, we proceed as follows.

Suppose we rotate the coordinate frame over an angle ϕ in counterclockwise direction, for some $0\leqslant\phi<2\pi$. We call the axes in this rotated coordinate system the x_ϕ-axis and the y_ϕ-axis. For a red imprecise point $r\in R$, let $f_r(\phi)$ denote the minimum x_ϕ-coordinate of any point in $\mathcal{I}_r$. Similarly, let $g_b(\phi)$ denote the maximum x_ϕ-coordinate of any point in $\mathcal{I}_b$; see Fig. 1. Now there is a possible separator that makes an angle $\phi+\pi/2$ with the positive x-axis if and only if $\max_{r\in R} f_r(\phi) < \min_{b\in B} g_b(\phi)$. Hence, to find whether there exists an angle ϕ that admits a possible separator we compute the upper envelope $E^+(F)$ of the set $F := \{f_r : r\in R\}$ and the lower envelope $E^-(G)$ of the set $G := \{g_b : b\in B\}$, and then check whether there is an angle ϕ where $E^+(F)$ lies below $E^-(G)$. Observe that we can also find all possible separators, by finding the region of points above $E^+(F)$ and below $E^-(G)$.

To compute $E^+(F)$ we proceed as follows. Note that any two functions f_r and $f_{r'}$ intersect at angles defined by a common outer tangent of $\mathcal{I}_r$ and $\mathcal{I}_{r'}$. We now split the domain $[0,2\pi)$ of ϕ into four sub-domains of length $\pi/2$. Within

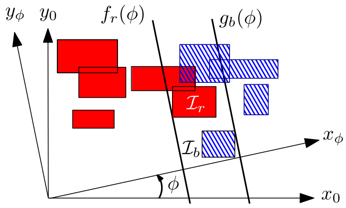

Fig. 1. The functions $f_r(\phi)$ and $g_b(\phi)$. Red imprecision regions are shaded. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

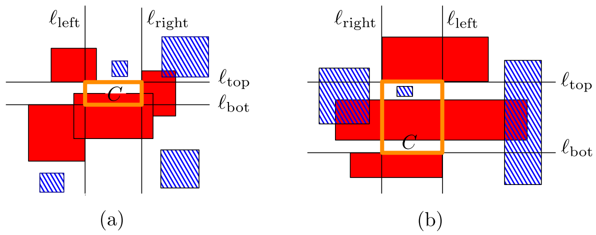

Fig. 2. The lines ℓ_{left}, ℓ_{right}, ℓ_{bot}, and ℓ_{top}. The red central region C is shown in dark-orange. (a) Illustrating the case where ℓ_{left} lies to the left of ℓ_{right} and ℓ_{bot} lies below ℓ_{top}. (b) Illustrating the case where ℓ_{left} lies to the right of ℓ_{right} and ℓ_{bot} lies below ℓ_{top}. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

each sub-domain the vertex of a rectangle $\mathcal{I}_r$ that determines f_r is fixed. Hence, within a sub-domain any two functions f_r and $f_{r'}$ intersect at most once, namely at the angle determined by the line through the two relevant vertices of $\mathcal{I}_r$ and $\mathcal{I}_{r'}$. Hence, the complexity of $E^+(F)$ is $O(n)$ and it can be computed in $O(n \log n)$ time [13]. The same is true for $E^-(G)$.

We conclude that deciding whether a possible linear separator exists (and, if so, computing one) can be done in $O(n \log n)$ time and $O(n)$ storage. Note that the fact that $E^+(F)$ and $E^-(G)$ have complexity $O(n)$ implies that the region of points above $E^+(F)$ and below $E^-(G)$ also has complexity $O(n)$. Thus we can actually compute (a representation of) all possible separators in $O(n \log n)$ time.

We now turn our attention to possible axis-parallel rectangular separators, where we assume without loss of generality that all red points should be inside and all blue points outside. Hence, we are looking for a rectangle σ such that no red imprecision region $\mathcal{I}_r$ is completely outside σ and no blue imprecision region is completely inside σ.

Consider all right edges of the red imprecision regions, and let ℓ_{left} be the vertical line through the leftmost of these edges. Clearly, any possible separator σ must have its left edge to the left of this line. Define ℓ_{right} as the vertical line through the rightmost of the left edges of the red imprecision regions, ℓ_{bot} as the lowest horizontal line through the top edges of the red imprecision regions, and ℓ_{top} as the highest horizontal line through the bottom edges of the red imprecision regions. We denote by C the rectangular closed region enclosed by these four lines and we call it the *red central region*; see Fig. 2. There are now several cases, depending on the relative positions of the lines ℓ_{left}, ℓ_{right}, ℓ_{bot}, and ℓ_{top}. For a separator σ we denote by $int(\sigma)$ its interior. For each of the four cases below we will define a *test set* $\mathcal{C}$ that helps us to decide if a rectangle σ is a possible separator.

(a) If ℓ_{left} lies to the left of (or on) ℓ_{right} and ℓ_{bot} lies below (or on) ℓ_{top}—Fig. 2(a)—then we say that C is a red central region of type (a). Define $\mathcal{C} := \{C\}$.
(b) If ℓ_{left} lies to the right of ℓ_{right} and ℓ_{bot} lies below (or on) ℓ_{top}—Fig. 2(b)—then we say that C is a red central region of type (b). In this case we define $\mathcal{C}$ to be the set of all vertical segments in C that connect ℓ_{bot} to ℓ_{top}.
(c) If ℓ_{left} lies to the left of (or on) ℓ_{right} and ℓ_{bot} lies above ℓ_{top} then we say that C is a red central region of type (c). In this case we define $\mathcal{C}$ to be the set of all horizontal segments in C that connect ℓ_{left} to ℓ_{right}.
(d) If ℓ_{left} lies to the right of ℓ_{right} and ℓ_{bot} lies above ℓ_{top} then we say that C is a red central region of type (d). In this case we define $\mathcal{C}$ to be the set of all points of C.

Now it is easy to see that rectangle σ is a possible separator if and only if (i) there exists an object $o \in \mathcal{C}$ such that $o \subset int(\sigma)$, and (ii) each blue region has a non-empty intersection with the complement of σ. As a consequence, for an input instance of red and blue imprecise points, we can decide if a possible separator exists in $O(n)$ time and storage. Indeed, if the red central region C is of type (a) we just need to check if each blue region intersects the complement of C, and if C is of type (b)–(d) then a possible separator always exists. Theorem 1 summarizes the results on certain and possible separators.

Theorem 1. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an imprecision region that is an axis-parallel rectangle. For linear separators, we can decide in $O(n)$ time and storage whether a certain separator exists for $R \cup B$ and in $O(n \log n)$*

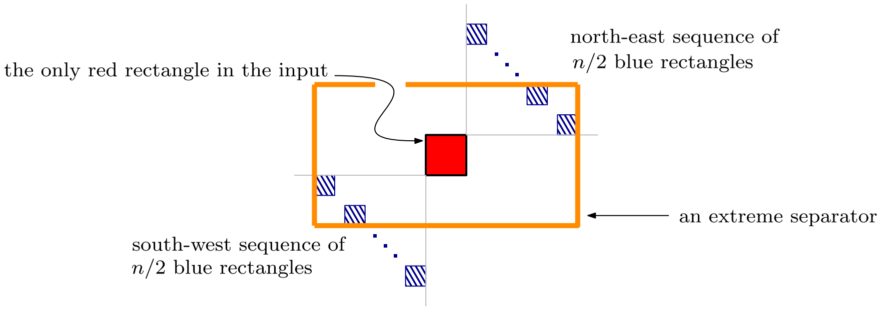

Fig. 3. Lower-bound construction for the number of extreme separators, consisting of a single red rectangle and two sequences of $n/2$ blue rectangles. Each pair of two consecutive north-east and two consecutive south-west blue region defines an extreme separator; the figure shows one such separator. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

time and $O(n)$ storage whether a possible separator exists. For axis-parallel rectangular separators, both problems can be solved in $O(n)$ time and storage.

If the imprecision regions are horizontal segments—there is only imprecision in the x-coordinates—then finding a possible linear separator can be done in $O(n)$ time by linear programming. Note that the results on certain separators as well as the results on possible separators do not need the probability distribution within the uncertainty regions to be uniform.

2.2. Computing all extreme separators

Above we showed how to decide if the sets R and B of red and blue imprecision regions admit a possible separator. Recall that for linear separators our technique can also report all possible separators. In this section we show how to compute a concise representation of the (possibly infinite) set of all possible rectangular separators. To this end, we call an axis-parallel rectangle σ an *extreme rectangular separator* if and only if the following conditions hold:

(i) σ is not a possible separator, and
(ii) any $\sigma' \subseteq int(\sigma)$ for which there exists an object $o \in \mathcal{C}$ with $o \subseteq int(\sigma')$ is a possible separator, where $\mathcal{C}$ is the test set defined on page 27, and
(iii) no axis-parallel rectangle $\sigma'' \subset \sigma$ satisfies conditions (i) and (ii).

In order to avoid unbounded separators we assume all the imprecision regions are inside a large axis-parallel rectangle $\mathcal{B}$ and we only consider extreme and possible separators that lie inside $\mathcal{B}$. Now it is easy to verify that the set of all extreme rectangular separators captures the set of all possible rectangular separators. More precisely,

Observation 2. *Let X be the set of all extreme rectangular separators. Then $\bigcup_{\sigma \in X} \Sigma(\sigma)$ is exactly the set of all possible separators, where $\Sigma(\sigma)$ denotes the set of all separators σ' satisfying the condition (ii) in the definition of an extreme rectangular separator.*

Following Observation 2, we now turn our attention to computing the set of all extreme rectangular separators. We use $x(\cdot)$ to denote the x-coordinate of a point or a vertical edge, and $y(\cdot)$ to denote the y-coordinate of a point or a horizontal edge.

First we prove a bound on the total number of extreme separators.

Theorem 3. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an axis-parallel rectangular imprecision region. Then the number of extreme rectangular separators for the set $R \cup B$ is $O(n^2)$ and this bound is tight in the worst case.*

Proof. Fig. 3 shows an instance for which there are $\Omega(n^2)$ extreme separators, thus proving the lower bound. Next we prove the upper bound.

To prove the upper bound, we first observe that the left edge of any extreme separator σ must contain the left edge of a blue rectangle (contained in the separator) or the left edge of $\mathcal{B}$, otherwise σ is not extreme. Now fix a left edge e_{left} of a blue rectangle, and let $S(e_{\text{left}})$ be the set of all extreme separators whose left edge contains e_{left}. We claim that $|S(e_{\text{left}})| \leqslant 2n$, which implies that the total number of extreme separators is $O(n^2)$. To prove the claim we will charge each separator to a top or bottom edge of a blue region in such a way that each top edge and each bottom edge is charged at most once. Our charging scheme is as follows.

Let $\sigma \in S(e_{\text{left}})$ be an extreme separator. Each of its four edges contains an edge e of a blue rectangle r_b such that r_b is fully contained in the interior of σ. (If not, then σ cannot be extreme.) Let e_{top} and e_{bot} be the edges contained in the top and bottom edges of σ, respectively. If the x-coordinate of the right endpoint of e_{top} is greater than the x-coordinate of the

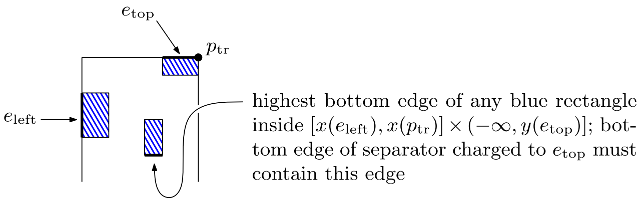

Fig. 4. Illustration to show that $e_{\rm top}$ is charged at most once. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

right endpoint of $e_{\rm bot}$, then we charge σ to $e_{\rm top}$; otherwise we charge σ to $e_{\rm bot}$. Now any edge is charged at most once. Indeed, consider a separator σ charged to a top edge $e_{\rm top}$, and let $p_{\rm tr}$ be the right endpoint of $e_{\rm top}$. Then the y-coordinate of the bottom edge of such a separator is uniquely determined: it can be obtained by taking a separator with left edge containing $e_{\rm left}$, top edge containing $e_{\rm top}$, right edge whose top endpoint is $p_{\rm tr}$, and then moving its bottom edge down as far as possible. In other words, the y-coordinate is given by the highest bottom edge of any blue rectangle r_b such that r_b lies completely inside the range $[x(e_{\rm left}), x(p_{\rm tr})] \times (-\infty, y(e_{\rm top})]$; see Fig. 4. Moreover, once the left, top and bottom edges of the separator are fixed, there can only be one position for the right edge. Thus there can indeed be at most one separator charged to $e_{\rm top}$. Similarly, there can be at most one separator charged to any bottom edge $e_{\rm bot}$. This proves our claim that $|S(e_{\rm left})| \leqslant 2n$. □

Let $S := R \cup B$ be a set of n red and blue imprecise points each with an axis-parallel imprecision region. Next we describe how to find all extreme rectangular separators for S in $O(n^2 \log n)$ time. The algorithm is an adaption of the upper-bound proof argument in Theorem 3.

Let $e_{\rm left}$ be a left edge of a blue region. We will show how to report the set $S(e_{\rm left})$ of all extreme separators whose left edge contains $e_{\rm left}$ in $O(n \log n)$ time. We do so in two rounds. In the first round we report the extreme separators σ such that the x-coordinate of the right endpoint of $e_{\rm top}(\sigma)$ is greater than the x-coordinate of the right endpoint of $e_{\rm bot}(\sigma)$, where $e_{\rm top}(\sigma)$ and $e_{\rm bot}(\sigma)$ are the edges contained in the top and bottom edges of σ, respectively. We call this a *type (i) separator*. In the second round we then report in a similar manner all extreme *type (ii) separators*, which are the separators such that the x-coordinate of the right endpoint of $e_{\rm top}(\sigma)$ is less than or equal to that of $e_{\rm bot}(\sigma)$. Next we describe the first round of the algorithm.

Recall from Section 2.1 that the red rectangles together define a red central region C such that at least one element of its corresponding test set $\mathcal{C}$ must be contained in any possible separator. Thus we may assume that $e_{\rm left}$ lies to the left of region C if C is of type (a) or (c), and to the left of the right edge of region C if C is of type (b) or (d). See Section 2.1 for different types of red central regions. We now proceed as follows. Let $\ell(e_{\rm left})$ be the vertical line containing $e_{\rm left}$.

1. Remove all blue regions that intersect or are to the left of $\ell(e_{\rm left})$. No matter how the top, bottom, and right edge of the separator σ are chosen, these regions cannot be fully contained in the interior of a separator σ whose left edge contains $e_{\rm left}$, so they can be ignored.
2. Let V be the set of bottom-right vertices of the remaining blue regions.
   - Construct a semi-dynamic (deletion-only) data structure $\mathcal{D}_{\rm x}$ on V for the following type of queries: given a query x-range $(-\infty, x)$, report the top-most point whose x-coordinate lies inside the range.
   - Construct a semi-dynamic (deletion-only) data structure $\mathcal{D}_{\rm y}$ on V for the following type of queries: given a y-range $(y, +\infty)$, report the left-most point whose y-coordinate lies inside the range.
3. Perform a plane-sweep with a horizontal sweep line moving downward. The sweep line will halt at every blue top edge $e_{\rm top}$, and then report the type (i) separator for $e_{\rm top}$ (if it exists). This is done as follows.
   When we halt at $e_{\rm top}$ we first remove from $\mathcal{D}_{\rm x}$ and $\mathcal{D}_{\rm y}$ the bottom-right vertex of the blue region corresponding to $e_{\rm top}$. Next we query $\mathcal{D}_{\rm x}$ to find the top-most point with x-coordinate in the range $(-\infty, x(p_{\rm tr}))$, where $p_{\rm tr}$ is the right endpoint of $e_{\rm top}$. Let the vertex q be the answer to the query. (If the query range is empty, then we define $y(q)$ to be the y-coordinate of the bottom edge of the rectangle $\mathcal{B}$.) Note that the blue rectangles whose bottom-right vertices are still stored in $\mathcal{D}_{\rm x}$ and $\mathcal{D}_{\rm y}$ are exactly the ones that are contained in $(x(e_{\rm left}), +\infty) \times (-\infty, y(e_{\rm top}))$, since any rectangle intersecting or to the left of $\ell(e_{\rm left})$ has been removed in Step 1, and any rectangle intersecting or above the line containing $e_{\rm top}$ has already been removed during the sweep. Similar to the charging argument in the proof of Theorem 3, we thus conclude that the type (i) separator containing $e_{\rm top}$ in its top edge (if it exists) must have a bottom edge with y-coordinate $y(q)$.
   Assuming the red central region C is of type (a) or (b), if $y(q)$ is greater than the y-coordinate of the bottom edge of region C then $e_{\rm top}$ does not define a type (i) separator with $e_{\rm left}$. Similarly when the red central region C is of type (c) or (d), if $y(q)$ is greater than the y-coordinate of the top edge of region C then $e_{\rm top}$ does not define a type (i) separator with $e_{\rm left}$. In either of the cases we are done with $e_{\rm top}$ and we continue with next horizontal blue top-edge. Otherwise, we perform a query in $\mathcal{D}_{\rm y}$ to find the leftmost bottom-right vertex in the range $(y(q), +\infty)$. Let q' be the answer to the query. If $x(q')$ is less than the x-coordinate of the right edge of region C in the case of a red central region of type (a)

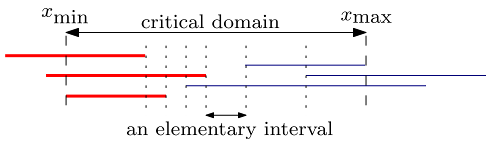

Fig. 5. Illustration of the critical domain and its elementary intervals. Red imprecision segments are thicker than blue imprecision segments. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

or (c), and if $x(q')$ is less than the x-coordinate of the left edge of region C in the case of a red central region of type (b) or (d), then e_{top} does not define a type (i) separator with e_{left}, and we continue the sweep. Otherwise, before continuing the sweep we first report the separator with left edge containing e_{left}, top edge containing e_{top}, bottom edge through q, and right edge through q'.

Theorem 4. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an axis-parallel rectangular imprecision region. Then we can report all extreme separators for the set $R \cup B$ in $O(n^2 \log n)$ time.*

Proof. The correctness of our algorithm follows from the discussion above. To prove the time bound we note that the data structures $\mathcal{D}_x$ and $\mathcal{D}_y$ can be implemented in a standard manner using augmented search trees, giving $O(\log n)$ query and deletion time. Thus the sweep for a fixed e_{left} runs in $O(n \log n)$ time, giving an overall time bound of $O(n^2 \log n)$. $\square$

2.3. Most likely separators

Finding most likely separators is considerably harder than finding possible separators. We study the 1-dimensional version of the problem, where the imprecision regions are intervals on the real line and a linear separator is a point. Suppose we are interested in separators that have all red points to the left and all blue points to the right.

For a point $x \in \mathbb{R}$ define $F(x) := \Pr[x \text{ is a separator}]$. For a red point r define lengthL(r, x) as the length of the part of $\mathcal{I}_r$ lying to the left of x, and for a blue point b define lengthR(b, x) as the length of the part of $\mathcal{I}_b$ lying to the right of x. Obviously the probability that a red point r lies to the correct side of x is equal to $f_r(x) := \text{lengthL}(r, x)/\text{length}(\mathcal{I}_r)$, and the probability that a blue point b lies to the correct side is $f_b(x) := \text{lengthR}(b, x)/\text{length}(\mathcal{I}_b)$. Hence,

$$F(x) = \prod_{r \in R} f_r(x) \cdot \prod_{b \in B} f_b(x).$$

Let $x_{\min}$ be the rightmost left endpoint of the red imprecision regions, and let $x_{\max}$ be the leftmost right endpoint of the blue imprecision regions. If $x_{\min} \geqslant x_{\max}$ then no separator exists, so assume $x_{\min} < x_{\max}$. Note that $F(x)$ is non-zero exactly on the interval $(x_{\min}, x_{\max})$, which we call the *critical domain* of F. The endpoints of the imprecision regions inside the critical domain partition it into *elementary intervals* as depicted in Fig. 5. Over each such elementary interval, the function $F(x)$ is a polynomial whose degree is upper bounded by the number of imprecision regions containing that elementary interval. We now prove that $F(x)$ is *unimodal* over its critical domain, which means that it has a single local maximum there.

Lemma 5. *$F(x)$ is unimodal over its critical domain.*

Proof. We first prove that $F(x)$ is unimodal in the interior of each elementary interval $I = [x_1, x_2]$, where we assume without loss of generality that $x_1 = 0$. For any red point r with $I \cap \mathcal{I}_r = \emptyset$ we have $f_r(x) = 1$ for $x \in I$. (We cannot have $f_r(x) = 0$ as I is part of the critical domain.) When $I \subseteq \mathcal{I}_r$, we have $f_r(x) = (C_r + x)/\text{length}(\mathcal{I}_r)$ for a constant C_r (which is the length of the part of $\mathcal{I}_r$ lying to the left of I). Similarly, for a blue point b for which $I \subseteq \mathcal{I}_b$ we have $f_b(x) = (C'_b - x)/\text{length}(\mathcal{I}_b)$ for a constant C'_b (which is the length of the part of $\mathcal{I}_b$ lying to the right of the left endpoint of I). Note that we must have $x \leqslant C'_b$ within I. Hence, if $R(I)$ and $B(I)$ are the sets of red and blue points whose imprecision regions cover I, then for $x \in I$

$$F(x) = C \cdot \prod_{r \in R(I)} (C_r + x) \cdot \prod_{b \in B(I)} (C'_b - x), \tag{1}$$

where $C = 1/(\prod_{r \in R(I)} \text{length}(\mathcal{I}_r) \cdot \prod_{b \in B(I)} \text{length}(\mathcal{I}_b))$. Thus

$$F'(x) = C \cdot F(x) \cdot \left(\sum_{r \in R(I)} \frac{1}{C_r + x} - \sum_{b \in B(I)} \frac{1}{C'_b - x} \right), \tag{2}$$

where F' denotes the first derivative of F. Note that $F(x) > 0$ for $x \in I$, all terms $1/(C_r + x)$ and $1/(C'_b - x)$ are positive, the sum $\sum_{r \in R(I)} 1/(C_r + x)$ is strictly decreasing while the sum $\sum_{b \in B(I)} 1/(C'_b - x)$ is strictly increasing. Hence, $F'(x) = 0$ at most once inside I or $F'(x) = 0$ everywhere inside I. (The latter occurs when $R(I) \cup B(I) = \emptyset$, which happens for at most one elementary interval.) Thus $F(x)$ is unimodal inside I.

To extend the analysis to the entire critical domain, we consider two consecutive elementary intervals I_1 and I_2. Let x^* be the right endpoint of I_1 (which is also the left endpoint of I_2). Denote the left and right derivative at x^* by $(F')^-(x^*)$ and $(F')^+(x^*)$, respectively. We claim that $(F')^-(x^*) > (F')^+(x^*)$. Observe that $R(I_1) \supseteq R(I_2)$. Indeed, a red imprecision region cannot start at x^* since then $F(x) = 0$ for $x \in I_1$. Similarly, $B(I_1) \subseteq B(I_2)$. From Equation (2) we now see that $(F')^-(x^*) > (F')^+(x^*)$. Together with the unimodality inside each elementary interval, this means $F(x)$ is unimodal over the entire critical domain. □

Lemma 5 allows us to perform a binary search over the critical domain to find the elementary interval I^* containing the most likely separator. At each step of the binary search, we need to evaluate $F(x)$ at a given x, which takes $O(n)$ time. Hence, I^* can be found in $O(n \log n)$ time in total. Unfortunately, the most likely separator X^* is not necessarily one of the endpoints of I^*. (For instance, when R consists of a single imprecision region $[0, 1]$ and B consists of a single imprecision region $[0.5, 1.5]$ then the most likely separator is the point $x = 0.75$.) Moreover, within I^* the function $F(x)$ is a polynomial of possibly very high degree. Hence, we may have to resort to numerical methods to approximate its maximum.

Theorem 6. *Let $R \cup B$ be a bichromatic set of n imprecise points on the real line, each with an imprecision region that is an interval. Then we can locate in $O(n \log n)$ time and $O(n)$ storage the elementary interval that contains the most likely separator.*

3. Weak separability

We now turn our attention to the case where we allow some of the points to be misclassified. The goal is then to find a *maximal separator*, that is, a separator that is expected to correctly classify the maximum number of points.

3.1. Weak separability by a line

For a directed line ℓ, let ℓ^- denote the halfplane to the left of ℓ and ℓ^+ the halfplane to the right of ℓ. We want to find a line ℓ that maximizes $G(\ell)$, which is defined as the expected number of red points in ℓ^- plus the expected number of blue points in ℓ^+. For a red point r we define $g_r^-(\ell)$ to be the fraction of the area of $\mathcal{I}_r$ lying to the left of ℓ, and for a blue point b we define $g_b^+(\ell)$ to be the fraction of the area of $\mathcal{I}_b$ to the right of ℓ. Hence, $g_r^-(\ell)$ and $g_b^+(\ell)$ give the probability that r and b are classified correctly, respectively, so

$$G(\ell) = \sum_{r \in R} g_r^-(\ell) + \sum_{b \in B} g_b^+(\ell). \tag{3}$$

To find the maximal separator, we dualize the corners of the imprecision regions, giving us a set L of $4n$ lines in dual space. With a slight abuse of notation we use $G(p)$, for a point p in dual space, to denote the value $G(\ell_p)$ of the line ℓ_p in primal space whose dual is p. Let G_C denote the function G restricted to a cell C of the arrangement $\mathcal{A}(L)$. For two neighboring cells C and C' we can obtain $G_{C'}$ from G_C by adding, subtracting or modifying one of the terms in (3). Let e be the edge of $\mathcal{A}(L)$ that is shared between the cells C and C'. Assume that e belongs to the line $\ell \in L$. Then according to whether ℓ dualizes to a red point or a blue point (in the primal plane) two cases occur. We explain the former case; the latter one can be handled similarly. Thus, assume that ℓ dualizes to a red point r. Now if the points in C dualize to lines above r and above the other three corners of the corresponding red imprecision region, then a term (related to the red imprecision region with corner r) is added to G_C to obtain $G_{C'}$. If the points in C dualize to lines above r but not above all the other three corners of the corresponding imprecision region, then a term is modified in G_C to obtain $G_{C'}$. Similarly, if the points in C dualize to lines below r then a term is either subtracted from or modified in G_C to obtain $G_{C'}$. Hence, we can compute a maximal separator by constructing the arrangement $\mathcal{A}(L)$ in $O(n^2)$ time, traversing the dual graph of the arrangement while maintaining the function G, and computing the maximum value of G in each cell C. Within each cell, G is a polynomial of degree two whose derivative helps to find the maximum. (Since the maximum inside C can be computed in $O(n_C)$ time, where n_C is the complexity of C, this takes $O(n^2)$ time in total.) We can improve the storage requirements of the algorithm by not computing the entire arrangement before we start the traversal, but by computing $\mathcal{A}(L)$ using topological sweep [9]. Besides the usual information we need to maintain for the sweep, we then also maintain the function G_C for each cell C immediately to the left of the sweep line. This way the maximal separator can be found using only $O(n)$ storage.

Theorem 7. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an axis-parallel rectangular imprecision region. We can compute a maximal linear separator for $R \cup B$ in $O(n^2)$ time using $O(n)$ storage.*

3.2. Weak separability by a rectangle

We now turn our attention to the problem of finding an axis-parallel rectangular separator σ that maximizes the sum of the expected number of red points inside σ and the expected number of blue points outside σ. This is equivalent to

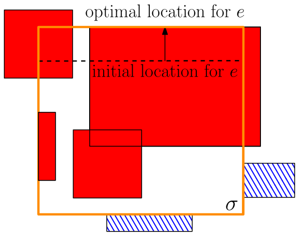

Fig. 6. There must be a maximal separator all of whose edges overlap at least partially with an edge of an imprecision region. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

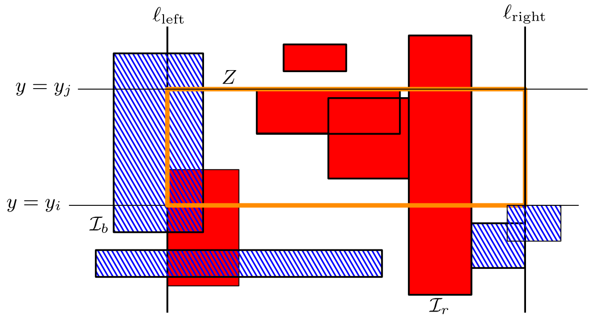

Fig. 7. Situation for a recursive call. The regions $\mathcal{I}_r$ and $\mathcal{I}_b$ should be taken into account inside Z even though they do not have a horizontal edge inside Z. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

maximizing $G(\sigma) := \sum_{r\in R} g_r^-(\sigma) + \sum_{b\in B} g_b^+(\sigma)$, where $g_r^-(\sigma)$ and $g_b^+(\sigma)$ denote the fractions of $\mathcal{I}_r$ and $\mathcal{I}_b$ that are covered by the interior and exterior of σ, respectively.

We first observe that there must be a maximal separator such that all of its edges overlap at least partially with an edge of an imprecision region. Indeed, if we keep three edges of a separator σ fixed and move the fourth edge e, as illustrated in Fig. 6, then $G(\sigma)$ changes linearly until we hit an edge of an imprecision region. Hence, there is a direction into which we can move e—either growing or shrinking σ—such that $G(\sigma)$ does not decrease until we hit an edge. This observation implies that there are only $O(n^4)$ candidates for the maximal separator. However, we can still compute a maximal separator in $O(n^3 \log n)$ time, as shown next.

Pick two vertical edges of imprecision regions. Let ℓ_{left} and ℓ_{right} be the vertical lines containing these edges, with ℓ_{left} lying to the left of ℓ_{right}. We will compute the maximal rectangular separator whose left and right edges are restricted to be contained in ℓ_{left} and ℓ_{right}, respectively, by a divide-and-conquer algorithm. Let $y_1, \ldots, y_m$ be the y-coordinates of the horizontal edges of the imprecision regions that lie at least partially inside the strip defined by ℓ_{left} and ℓ_{right}. We can assume these y-coordinates are sorted in increasing order. Let $t := \lceil m/2 \rceil$, and let ℓ_{mid} be the line $y = y_t$. The idea is to compute the best separators above and below ℓ_{mid} recursively, then compute the best separator intersecting ℓ_{mid}, and then take the best of the three separators. Computing the best separator intersecting ℓ_{mid} seems easy: We just compute the best separator whose bottom edge is contained in ℓ_{mid} by scanning the possible y-coordinates for the top edge in the order $y_{t+1}, \ldots, y_m$, do the same for the best separator whose top edge is contained in ℓ_{mid} (this time scanning downward over $y_{t-1}, \ldots, y_1$) and take the union of the two separators found. However, in the recursive call we may have to take into account those imprecision regions whose top and bottom edges fall outside the y-range corresponding to the recursive call; see Fig. 7. This will be problematic for the running time. Hence, we refine our algorithm as follows.

In a generic recursive call we are given a rectangle Z bounded from the left by ℓ_{left}, from the right by ℓ_{right}, from below by the line $y = y_i$ for some $1 \leqslant i < m$ (initially $i = 1$), and from the top by the line $y = y_j$, for some $i < j \leqslant m$ (initially $j = m$). We also have a sorted list of all y-coordinates $y_i, \ldots, y_j$ of the horizontal edges of the imprecision regions that intersect Z, with for each y-coordinate a pointer to the imprecision region that generated it. Our goal is to compute the best separator contained in Z whose left and right edges are contained in ℓ_{left} and ℓ_{right}, respectively, and whose bottom and top edges have y-coordinates chosen from the list. To this end we also need some information to deal with the imprecision regions that intersect Z but do not have a horizontal edge intersecting Z.

Consider such a red imprecision region $\mathcal{I}_r$, and consider a separator $\sigma(y) := [x_{\mathrm{left}}, x_{\mathrm{right}}] \times [y_i, y]$, where x_{left} and x_{right} are the x-coordinates of ℓ_{left} and ℓ_{right}, respectively. Define $f_r(y)$ to be the fraction of $\mathcal{I}_r$ inside $\sigma(y)$. Note that $f_r(y)$ is a linear function. Also note that the fraction of $\mathcal{I}_r$ inside a separator $[x_{\mathrm{left}}, x_{\mathrm{right}}] \times [y', y]$ is given by $f_r(y) - f_r(y')$. For a blue imprecision region $\mathcal{I}_b$ we define $f_b(y)$ similarly, except this time we use the fraction of $\mathcal{I}_b$ outside $\sigma(y)$. The extra information we need in the recursive call with region Z is the linear function $F_Z := \sum_{r\in R} f_r + \sum_{b\in B} f_b$.

It remains to describe how to handle the recursive call with rectangle Z. We split Z into two rectangles Z_1 and Z_2 at y-coordinate y_t, where $t := \lceil (i+j)/2 \rceil$ is the median y-coordinate of the horizontal edges in Z and Z_1 is the lower rectangle.

Next we compute the functions F_{Z_1} and F_{Z_2} that we have to pass on to the recursive calls for Z_1 and Z_2. The function F_{Z_1} can be computed in linear time as follows. We first determine all imprecision regions that have a horizontal edge in Z but not in Z_1. We compute the functions f_r (resp. f_b) for all such red (resp. blue) imprecision regions. We add all these functions and then add the function F, which represents the contributions of the imprecision regions that already span Z. For F_{Z_2} the computations are similar, except that we should subtract $F(y_t)$, since F was defined for separators whose bottom edge has y-coordinate y_i while F_{Z_2} is defined for separators whose bottom edge has y-coordinate y_t. Thus both F_{Z_1} and F_{Z_2} can be computed in linear time.

We now do recursive calls on Z_1 with function F_{Z_1} and on Z_2 with function F_{Z_2}, giving us two candidate separators. After the recursive calls, we have to find the best separator that intersects $y = y_t$. To this end, we first compute the best separator σ_1^* contained in Z and of the form $[x_{\text{left}}, x_{\text{right}}] \times [y, y_t]$ and the best separator σ_2^* contained in Z and of the form $[x_{\text{left}}, x_{\text{right}}] \times [y_t, y]$. Both can be computed by scanning edges of the imprecision regions in order—for the former separator we scan downwards from y_t, for the latter we scan upwards from y_t—and maintaining the expected number of correctly classified points. While we scan, we use the function F to account for the contribution of the imprecision regions without a horizontal edge inside Z. This way the scans can be implemented so that they run in linear time. The best separator intersecting $y = y_t$ is now given by $\sigma_1^* \cup \sigma_2^*$.

We conclude that we need $O(n)$ time to handle Z, plus the time needed for the calls on Z_1 and Z_2, leading to a total time of $O(n \log n)$ to find the best separator whose left and right edges are contained in the lines ℓ_{left} and ℓ_{right}. The overall time for the algorithm is therefore $O(n^3 \log n)$, and the storage is easily seen to be $O(n)$.

Theorem 8. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an imprecision region that is an axis-parallel rectangle. We can compute a maximal rectangular separator for $R \cup B$ in $O(n^3 \log n)$ time and $O(n)$ storage.*

Horizontal segments as imprecision regions. We can improve the running time even further when the imprecision regions are horizontal unit-length segments rather than rectangles.[3] As in the case of rectangular imprecision regions, we only have to consider separators σ whose left and right edges pass through a vertex of an imprecision region. We will first consider a special case of the problem, where the maximal rectangular separator is required to intersect a given horizontal line. The solution to this problem will be used as a subroutine in a divide-and-conquer algorithm for the general problem.

The restricted problem. Let ℓ_{hor} be a given horizontal line that contains none of the input segments. We call a rectangular separator that intersects ℓ_{hor} a *restricted separator*. Our goal is to compute a restricted separator that maximizes the expected number of correctly classified points. As mentioned above, we only have to consider separators whose left edge passes through an endpoint of an imprecision region. Fix an endpoint v, and let ℓ_{vert} denote the vertical line through v. We further restrict our separator by requiring that its left edge is contained in ℓ_{vert}. We show how to compute such a maximal separator for ℓ_{vert} in $O(n\sqrt{n})$ time, leading to an algorithm for the restricted problem that runs in $O(n^2\sqrt{n})$ time.

Let $\mathcal{I}_1, \dots, \mathcal{I}_m$ be the parts of the imprecision regions in $R \cup B$ lying to the right of ℓ_{vert}, numbered from top to bottom. Let y_i denote the y-coordinate of $\mathcal{I}_i$, and let k be such that $\mathcal{I}_1, \dots, \mathcal{I}_k$ lie above ℓ_{hor} and $\mathcal{I}_{k+1}, \dots, \mathcal{I}_m$ lie below ℓ_{hor}. To simplify the notation, we assume that the imprecision regions $\mathcal{I}_1, \dots, \mathcal{I}_m$ all have distinct y-coordinates. For $0 \leqslant i \leqslant m$ we define y_i' to be an (arbitrary) y-coordinate in the range (y_{i+1}, y_i), where y_0 is the y-coordinate of the top edge of $\mathcal{B}$ and y_{m+1} is the y-coordinate of the bottom edge of $\mathcal{B}$. (If the y-coordinate of $\mathcal{I}_1, \dots, \mathcal{I}_m$ were not distinct, we would only pick y_i''s between consecutive but distinct values y_i and y_{i+1}.) Now for each $1 \leqslant i \leqslant k$ and $x > 0$, define $\sigma_i(x)$ to be the rectangle bounded from the left by ℓ_{vert}, bounded from below by ℓ_{hor}, bounded from above by the line $y = y_i'$, and bounded from the right by the vertical line at distance x from ℓ_{vert}. See Fig. 8.

For $k < i \leqslant m$ we define $\sigma_i(x)$ similarly, except that now $\sigma_i(x)$ is bounded from above by ℓ_{hor} and from below by $y = y_i'$. Finally, let $\sigma^*(x)$ denote the restricted separator whose left edge is contained in ℓ_{vert} and whose right edge lies at distance x from ℓ_{vert} for which the expected number of correctly classified points is maximized. Clearly $\sigma^*(x)$ is obtained by combining the best rectangle from the set $\{\sigma_i(x) : 1 \leqslant i \leqslant k\}$ with the best rectangle from $\{\sigma_i(x) : k < i \leqslant m\}$. Hence, $G(\sigma^*(x)) = \max_{1 \leqslant i \leqslant k} G(\sigma_i(x)) + \max_{k < i \leqslant m} G(\sigma_i(x))$.

To find the overall best rectangle, we need to find the maximum of $G(\sigma^*(x))$ over all $x > 0$. (In fact, we know that we only have to consider x-values corresponding to the endpoints of the imprecision regions. However, in our approach we do not restrict our attention to those values only.) Thus our strategy is to compute the upper envelopes of the sets of functions $\Gamma := \{G(\sigma_i(x)) : 1 \leqslant i \leqslant k\}$ and $\overline{\Gamma} := \{G(\sigma_i(x)) : k < i \leqslant m\}$. Once we have the upper envelopes, we can add them in linear time (in the sum of their complexities) to find the best restricted rectangular separator with left edge at ℓ_{vert}. Next we describe how to compute $\mathcal{E}(\Gamma)$, the upper envelope of Γ; computing $\mathcal{E}(\overline{\Gamma})$ can be done in a similar way.

[3] The results above are stated for rectangular imprecision regions, but it is easily checked that they can be extended to imprecision regions that are either rectangles, or horizontal segments, or vertical segments. The goal here is to improve these results when we only have horizontal unit-length segments as imprecision regions.

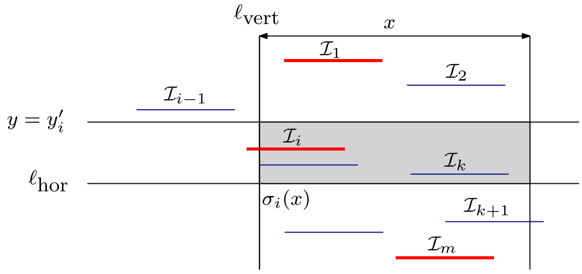

Fig. 8. The shaded area shows the rectangle $\sigma_i(x)$. Red imprecision segments are thicker than blue imprecision segments. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

To compute $\mathcal{E}(\Gamma)$ we use a divide-and-conquer algorithm. Define $\Gamma' := \{G(\sigma_i(x)) : 1 \leqslant i \leqslant t\}$ and $\Gamma'' := \{G(\sigma_i(x)) : t < i \leqslant k\}$, where $t := \lceil k/2 \rceil$. We will compute $\mathcal{E}(\Gamma')$ and $\mathcal{E}(\Gamma'')$ separately and merge the resulting envelopes to get $\mathcal{E}(\Gamma)$. Next we explain how to compute $\mathcal{E}(\Gamma')$ and $\mathcal{E}(\Gamma'')$.

Let $S := \{\mathcal{I}_1, \dots, \mathcal{I}_k\}$ denote the parts of the imprecision regions above ℓ_{hor} and to the right of ℓ_{vert}, and define $S' := \{\mathcal{I}_1, \dots, \mathcal{I}_t\}$ and $S'' := \{\mathcal{I}_{t+1}, \dots, \mathcal{I}_k\}$ to be the top half and bottom half of the set S of imprecision regions, respectively. The function values $G(\sigma_i(x))$ for $t < i \leqslant k$ only depend on S'', as all imprecision regions in S' are above any rectangle $\sigma_i(x)$ for $t < i \leqslant k$. Hence, we can compute $\mathcal{E}(\Gamma'')$ recursively by only considering S''. The function values $G(\sigma_i(x))$ for $1 \leqslant i \leqslant t$ depend on both S' and S''. However, each such $G(\sigma_i(x))$ can be obtained by computing $G(\sigma_i(x))$ with respect to S' and then adding $G(\sigma_{t+1}(x))$ to take S'' into account. Hence, we recursively compute $\mathcal{E}(\Gamma')$ with respect to S' and then we add $G(\sigma_{t+1})$ to the computed envelope to obtain the true envelope. Note that $G(\sigma_{t+1})$ can easily be computed in $O(n)$ time.

As mentioned, after computing $\mathcal{E}(\Gamma')$ and $\mathcal{E}(\Gamma'')$ as just described, we merge them to obtain $\mathcal{E}(\Gamma)$. Next we analyze $|\mathcal{E}(\Gamma)|$, the complexity of $\mathcal{E}(\Gamma)$.

Lemma 9. $|\mathcal{E}(\Gamma)| = O(n\sqrt{n})$

Proof. We partition the part of the plane to the right of ℓ_{vert} into $O(n)$ strips by drawing vertical lines through the endpoints of the imprecision regions. Now consider a function $G(\sigma_i(x))$. Since the imprecision regions are unit-length segments, the contribution of each red imprecision region $\mathcal{I}_r$ to $G(\sigma_i(x))$ is equal to the length of $\mathcal{I}_r$ inside $\sigma_i(x)$. Similarly, for a blue imprecision region $\mathcal{I}_b$, the contribution is its length outside $\sigma_i(x)$. This implies that $G(\sigma_i(x))$, which is the sum of all contributions, is linear within each strip. Moreover, the slope of $G(\sigma_i(x))$ is equal to the number of red imprecision regions inside the strip that are below y_i minus the number of such blue imprecision regions. Note that this implies that the slope of $G(\sigma_i(x))$ in adjacent strips differs by at most 1. (This assumes that all endpoints of the imprecision regions have distinct x-coordinates. When this is not the case, we can introduce a number of dummy strips of zero width at shared x-coordinates, and the argument still goes through.) Using the above observation we can now bound the complexity of the upper envelope $\mathcal{E}(\Gamma)$ using a charging scheme, as follows.

Number the strips from left to right, and consider the j-th strip. Let E_j be the set of edges of $\mathcal{E}(\Gamma)$ that lie in the j-th strip. We charge the $\sqrt{n}$ rightmost edges of E_j to the j-th strip, and we charge the remaining edges to the function $G(\sigma_i)$ contributing this edge. Obviously the total charge to the strips is $O(n\sqrt{n})$, so it remains to bound the number of times any $G(\sigma_i)$ can be charged. To bound this number we observe that within a strip the upper envelope is a convex chain, so the slopes of the edges on the envelope strictly increase from left to right. Since within each strip we have a collection of linear functions (see the discussion above), the upper envelope of the functions is a sequence of line segments with increasing slopes from left to right. Since all slopes are integers, this implies that the slope of any function $G(\sigma_i)$ that is charged in the j-th strip is at least $\sqrt{n}$ smaller than the slope of the function contributing the rightmost edge of the envelope in the strip. Because the slope of any function changes by at most 1 from one strip to the next, this implies that it will take at least $\sqrt{n}/2$ strips for $G(\sigma_i)$ to overtake the function contributing the rightmost edge in the j-th strip. Hence, $G(\sigma_i)$ is charged $O(\sqrt{n})$ times, which concludes the proof. □

Lemma 9 implies that the running time of our algorithm for the restricted problem satisfies $T(n) = 2T(n/2) + O(n) + O(n\sqrt{n})$, when we fix the left edge of the rectangular separator to be contained in a vertical line ℓ_{vert}. This solves to $O(n\sqrt{n})$. Since ℓ_{vert} can be chosen in $O(n)$ ways, the running time is $O(n^2\sqrt{n})$, and according to $|\mathcal{E}(\Gamma)|$ the storage is $O(n\sqrt{n})$.

Lemma 10. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an imprecision region that is a unit-length horizontal segment, and let ℓ_{hor} be a horizontal line. Then in $O(n^2\sqrt{n})$ time and $O(n\sqrt{n})$ storage we can compute a maximal rectangular separator for $R \cup B$ that is restricted to intersect ℓ_{hor}.*

The general problem. With the solution for the restricted problem at hand we can easily obtain a divide-and-conquer algorithm for the general problem, where the separator is not required to intersect a given line. To this end we partition the plane into two half-planes by a horizontal line ℓ_{hor}, each containing half of the segments (imprecision regions) from $R \cup B$. We then recursively compute the maximal rectangular separator lying below ℓ_{hor}, and the maximal rectangular separator lying above ℓ_{hor}. Finally, we compute the maximal rectangular separator intersecting ℓ_{hor}—this can be done in $O(n^2\sqrt{n})$ by Lemma 10—and we take the best of the three separators computed. The total running time $T(n)$ of our algorithm satisfies $T(n) = 2T(n/2) + O(n^2\sqrt{n})$, which solves to $T(n) = O(n^2\sqrt{n})$, and the storage required is $O(n\sqrt{n})$.

Theorem 11. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with an imprecision region that is a unit-length horizontal segment. We can compute a maximal rectangular separator for $R \cup B$ in $O(n^2\sqrt{n})$ time and using $O(n\sqrt{n})$ storage.*

3.3. Approximate weak separability

Our exact algorithms to compute maximal separators have at least quadratic running time both for linear and for rectangular separators. We now present a simple near-linear $(1-\varepsilon)$-approximation algorithm for computing maximal separators. The approach works for linear as well as rectangular separators and in fact, could also be used when the imprecision regions are disks, for instance.

Define $\mathcal{R}$ to be a set of ranges corresponding to the type of separator we are interested in: for the linear separability $\mathcal{R}$ is the set of all possible half-planes, for the rectangular separability problem $\mathcal{R}$ is the set of all possible rectangles in the plane. First we replace each imprecision region $\mathcal{I}$ with a point set $P_{\mathcal{I}}$ such that, for any range in $\mathcal{R}$, the fraction of points inside the range is a good approximation of the fraction of the area of $\mathcal{I}$ that is covered by the range. More precisely, for each imprecision region $\mathcal{I}$, we compute a point set $P_{\mathcal{I}} \subset \mathcal{I}$ whose geometric discrepancy with respect to $\mathcal{R}$ is at most $\delta_1 := \varepsilon/8$, that is, such that for any range $\rho \in \mathcal{R}$ we have $\left|\, \text{area}(\rho \cap \mathcal{I})/\text{area}(\mathcal{I}) - |\rho \cap P_{\mathcal{I}}|/|P_{\mathcal{I}}| \,\right| \leqslant \delta_1$. All $\mathcal{I} \in R \cup B$ use a point set of the same size and the points in each set $P_{\mathcal{I}}$ are assigned the same color as $\mathcal{I}$.

Let $P_R := \bigcup_{r\in R} P_{\mathcal{I}_r}$ and $P_B := \bigcup_{b\in B} P_{\mathcal{I}_b}$ be the red and the blue points sets generated for all imprecision regions. We reduce the size of P_R by computing a δ_2-approximation A_R of P_R with respect to $\mathcal{R}$, that is, a subset $A_R \subset P_R$ such that $\left|\, |\rho \cap A_R|/|A_R| - |\rho \cap P_R|/|P_R| \,\right| \leqslant \delta_2$ for any range $\rho \in \mathcal{R}$, where $\delta_2 := \varepsilon/8$. The size of P_B is reduced similarly, obtaining a subset $A_B \subset P_B$.

Finally, we compute a separator σ_{ALG} from the class we are interested in—either lines or rectangles—that maximizes

$$\frac{|\sigma^+_{\text{ALG}} \cap A_R|}{|A_R|} \cdot |P_R| + \frac{|\sigma^-_{\text{ALG}} \cap A_B|}{|A_B|} \cdot |P_B|$$

where σ^+_{ALG} and σ^-_{ALG} denote the parts of the plane that are to the left and right (for lines) or inside and outside (for rectangles) our separator. This is done in a brute-force manner, by checking all separators on the point set $A_R \cup A_B$ (of which there are $O(|A_R \cup A_B|^2)$ for linear separators and $O(|A_R \cup A_B|^4)$ for rectangular separators). The following lemma proves the approximation ratio of our algorithm.

Lemma 12. *Let σ_{ALG} be the separator computed by our algorithm. Then $G(\sigma_{\text{ALG}}) \geqslant (1-\varepsilon) \cdot \text{OPT}$.*

Proof. For a separator σ, let $G(\sigma)$ denote the expected number of correctly classified points. Let σ_{opt} denote the separator that maximizes the expected number of correctly classified points, and let $\text{OPT} := G(\sigma_{\text{opt}})$. To simplify notation, we denote the region enclosed by σ_{ALG} by σ^+_{ALG}, and its complement by σ^-_{ALG}. By definition,

$$G(\sigma_{\text{ALG}}) = \sum_{r\in R} \frac{\text{area}(\sigma^+_{\text{ALG}} \cap \mathcal{I}_r)}{\text{area}(\mathcal{I}_r)} + \sum_{b\in B} \frac{\text{area}(\sigma^-_{\text{ALG}} \cap \mathcal{I}_b)}{\text{area}(\mathcal{I}_b)}$$

Since our algorithm replaces each region $\mathcal{I}_p$, for $p \in R \cup B$, with a point set $P_{\mathcal{I}_p}$ of discrepancy of at most δ_1, we have for every red point r and blue point b

$$\left| \frac{\text{area}(\sigma^+_{\text{ALG}} \cap \mathcal{I}_r)}{\text{area}(\mathcal{I}_r)} - \frac{|\sigma^+_{\text{ALG}} \cap P_{\mathcal{I}_r}|}{|P_{\mathcal{I}_r}|} \right| \leqslant \delta_1$$

and

$$\left| \frac{\text{area}(\sigma^-_{\text{ALG}} \cap \mathcal{I}_b)}{\text{area}(\mathcal{I}_b)} - \frac{|\sigma^-_{\text{ALG}} \cap P_{\mathcal{I}_b}|}{|P_{\mathcal{I}_b}|} \right| \leqslant \delta_1.$$

Hence,

$$G(\sigma_{\mathrm{ALG}}) \geqslant \frac{1}{k}\sum_{r\in R}|\sigma^{+}_{\mathrm{ALG}} \cap P_{\mathcal{I}_r}| + \frac{1}{k}\sum_{b\in B}|\sigma^{-}_{\mathrm{ALG}} \cap P_{\mathcal{I}_b}| - \delta_1(|R|+|B|)$$
$$= \frac{1}{k}\cdot\left[\,|\sigma^{+}_{\mathrm{ALG}} \cap P_R| + |\sigma^{-}_{\mathrm{ALG}} \cap P_B|\,\right] - \delta_1(|R|+|B|)$$

where $k := |P_{\mathcal{I}_r}|$ is the number of points we need in the low-discrepancy point sets. (Recall that all $\mathcal{I} \in R \cup B$ use a point set of the same size.) Since A_R and A_B are δ_2-approximations of P_R and P_B, respectively, we have

$$\left|\frac{|\sigma^{+}_{\mathrm{ALG}} \cap P_R|}{|P_R|} - \frac{|\sigma^{+}_{\mathrm{ALG}} \cap A_R|}{|A_R|}\right| \leqslant \delta_2 \qquad \text{and} \qquad \left|\frac{|\sigma^{-}_{\mathrm{ALG}} \cap P_B|}{|P_B|} - \frac{|\sigma^{-}_{\mathrm{ALG}} \cap A_B|}{|A_B|}\right| \leqslant \delta_2.$$

Thus,

$$G(\sigma_{\mathrm{ALG}}) \geqslant \frac{1}{k}\cdot\left[\,|\sigma^{+}_{\mathrm{ALG}} \cap P_R| + |\sigma^{-}_{\mathrm{ALG}} \cap P_B|\,\right] - \delta_1(|R|+|B|)$$
$$\geqslant \frac{1}{k}\cdot\left[\frac{|\sigma^{+}_{\mathrm{ALG}} \cap A_R|}{|A_R|}\cdot|P_R| + \frac{|\sigma^{-}_{\mathrm{ALG}} \cap A_B|}{|A_B|}\cdot|P_B| - \delta_2(|P_R|+|P_B|)\right] - \delta_1(|R|+|B|)$$
$$= \frac{1}{k}\cdot\left[\frac{|\sigma^{+}_{\mathrm{ALG}} \cap A_R|}{|A_R|}\cdot|P_R| + \frac{|\sigma^{-}_{\mathrm{ALG}} \cap A_B|}{|A_B|}\cdot|P_B|\right] - (\delta_1+\delta_2)(|R|+|B|)$$
$$\geqslant \frac{1}{k}\cdot\left[\frac{|\sigma^{+}_{\mathrm{opt}} \cap A_R|}{|A_R|}\cdot|P_R| + \frac{|\sigma^{-}_{\mathrm{opt}} \cap A_B|}{|A_B|}\cdot|P_B|\right] - (\delta_1+\delta_2)(|R|+|B|)$$
$$\text{(as } \sigma_{\mathrm{ALG}} \text{ maximizes the expression between the square brackets)}$$
$$\geqslant \frac{1}{k}\cdot\left[\left(\frac{|\sigma^{+}_{\mathrm{opt}} \cap P_R|}{|P_R|} - \delta_2\right)\cdot|P_R| + \left(\frac{|\sigma^{-}_{\mathrm{opt}} \cap P_B|}{|P_B|} - \delta_2\right)\cdot|P_B|\right] - (\delta_1+\delta_2)(|R|+|B|)$$
$$= \frac{1}{k}\cdot\left[\,|\sigma^{+}_{\mathrm{opt}} \cap P_R| + |\sigma^{-}_{\mathrm{opt}} \cap P_B|\,\right] - (\delta_1+2\delta_2)(|R|+|B|)$$
$$= \frac{1}{k}\cdot\left[\,\sum_{r\in R}|\sigma^{+}_{\mathrm{opt}} \cap P_{I_r}| + \sum_{b\in B}|\sigma^{-}_{\mathrm{opt}} \cap P_{I_b}|\,\right] - (\delta_1+2\delta_2)(|R|+|B|)$$
$$= \sum_{r\in R}\frac{|\sigma^{+}_{\mathrm{opt}} \cap P_{\mathcal{I}_r}|}{|P_{\mathcal{I}_r}|} + \sum_{b\in B}\frac{|\sigma^{-}_{\mathrm{opt}} \cap P_{\mathcal{I}_b}|}{|P_{\mathcal{I}_b}|} - (\delta_1+2\delta_2)(|R|+|B|)$$
$$\geqslant \sum_{r\in R}\frac{\mathrm{area}(\sigma^{+}_{\mathrm{opt}} \cap \mathcal{I}_r)}{\mathrm{area}(\mathcal{I}_r)} + \sum_{b\in B}\frac{\mathrm{area}(\sigma^{-}_{\mathrm{opt}} \cap \mathcal{I}_b)}{\mathrm{area}(\mathcal{I}_b)} - 2(\delta_1+\delta_2)(|R|+|B|)$$
$$= \textsc{Opt} - 2(\delta_1+\delta_2)n$$

Since $\textsc{Opt} \geqslant n/2$ (we can always correctly classify the larger of the sets R and B) it suffices to have $\delta_1+\delta_2 \leqslant \varepsilon/4$ in order to achieve the claimed bound. □

Theorem 13. *Let $R \cup B$ be a bichromatic set of n imprecise points in the plane, each with a rectangular imprecision region. We can compute a $(1-\varepsilon)$-approximation of the maximal linear separator for $R \cup B$ in $O(\frac{n}{\varepsilon^6}\log^2\frac{1}{\varepsilon})$ time. A $(1-\varepsilon)$-approximation of the maximal rectangular separator for $R \cup B$ can also be computed in $O(\frac{n}{\varepsilon^9}\log^5\frac{1}{\varepsilon})$ time.*

Proof. We first prove the bounds for the case of rectangular separators. For each imprecision region $\mathcal{I}$, the algorithm generates a point set $P_{\mathcal{I}}$ of size $O((1/\delta_1)\log(1/\delta_1))$ whose discrepancy for axis-parallel boxes is at most δ_1. Note that point sets $P_{\mathcal{I}}$ have all the same size. The Van der Corput set has this property [21, Section 2.1]. In the second step, the algorithm computes δ_2-approximations for P_B and P_R using a result by Matoušek [22], which states that a $(1/r)$-approximation of size $O(r^2\log r)$ for a point set with m points can be computed in time $O(m(r^2\log r)^d)$, where d is the exponent of the so-called shatter function of the underlying range space. For rectangles we have $d=4$, so plugging in $m = O(n(1/\delta_1)\log(1/\delta_1))$ and $r = 1/\delta_2$ gives a δ_2-approximation of size $O((1/\delta_2)^2\log(1/\delta_2))$ in time $O(n(1/\delta_1)\log(1/\delta_1)(1/\delta_2)^8\log^4(1/\delta_2))$. Since the size of the δ_2-approximations is constant, the last step of the algorithm takes constant time. (For simplicity we ignore the additive term in the running time which only depends on ε and not on n.) Thus, the algorithm will run in $O((1/\delta_1)\log(1/\delta_1)n(1/\delta_2)^8\log^4(1/\delta_2))$ time for rectangular separators. Since $\delta_1 = \delta_2 = \varepsilon/8$, we get the claimed running time.

When we are looking for linear separators, we need to replace each imprecision region with a point set $P_{\mathcal{I}}$ that has discrepancy at most δ_1 for half-planes. For this we simply take a regular grid of $k \times k$ points inside $\mathcal{I}$. For a suitable value $k = O(1/\delta_1)$ the discrepancy of the grid is easily seen to be at most δ_1. (In fact, it is known that there exists a set of $O(1/\delta_1^{4/3})$ points whose half-plane discrepancy is at most δ_1 [21, Chapter 5]. Since the construction algorithm for this set is more complicated and randomized, we use the grid instead, at the cost of a slight increase in the dependency on ε in our final bound.) The rest of the algorithm is the same as the other case. Since the exponent of the shatter function for half-planes is 2, the algorithm will run in $O(n(1/\delta_1)^2(1/\delta_2)^4 \log^2(1/\delta_2))$ time for linear separators, which is $O(n(1/\varepsilon)^6 \log^2(1/\varepsilon))$. $\square$

4. Concluding remarks

We believe that the set of problems that we studied are only a first step in studying geometric separating problems in an imprecise context, and many interesting problems remain to be studied. An obvious direction is to study other types of imprecision regions, such as disks. Second, studying separability in higher dimensions, or with other types of separators, are interesting directions for further research. Furthermore, it would be interesting to find maximal separators when the probability distributions in the imprecision regions are not uniform but, say, Gaussian.

References

[1] P.K. Agarwal, S. Har-Peled, S. Suri, H. Yildiz, W. Zhang, Convex hulls under uncertainty, in: Proc. 22nd Europ. Symp. Alg., 2014, pp. 37–48.
[2] E.M. Arkin, F. Hurtado, J.S.B. Mitchell, C. Seara, S. Skiena, Some lower bounds on geometric separability problems, Int. J. Comput. Geom. Appl. 16 (1) (2006) 1–26.
[3] B. Aronov, D. Garijo, Y. Núñez Rodríguez, D. Rappaport, C. Seara, J. Urrutia, Minimizing the error of linear separators on linearly inseparable data, Discrete Appl. Math. 160 (10–11) (2012) 1441–1452.
[4] S. Bereg, J.M. Díaz-Báñez, D. Lara, P. Pérez-Lantero, C. Seara, J. Urrutia, On the coarseness of bicolored point sets, Comput. Geom. Theory Appl. 46 (1) (2013) 65–77.
[5] M. de Berg, E. Mumford, M. Roeloffzen, Finding structures on imprecise points, in: Abstracts 26th Europ. Workshop Comput. Geom., 2010, pp. 85–88.
[6] T.M. Chan, Low-dimensional linear programming with violations, SIAM J. Comput. 34 (2005) 879–893.
[7] C. Cortés, J.M. Díaz-Báñez, P. Pérez-Lantero, C. Seara, J. Urrutia, I. Ventura, Bichromatic separability with two boxes: a general approach, J. Algebra 64 (2009) 79–88.
[8] M. Davoodi, A. Mohades, F. Sheikhi, P. Khanteimouri, Data imprecision under λ-geometry, Inf. Sci. 295 (2015) 126–144.
[9] H. Edelsbrunner, L.J. Guibas, Topologically sweeping an arrangement, J. Comput. Syst. Sci. 38 (1) (1989) 165–194.
[10] H. Edelsbrunner, F.P. Preparata, Minimum polygonal separation, Inf. Comput. 77 (1988) 218–232.
[11] H. Everett, J.M. Robert, M. van Kreveld, An optimal algorithm for computing ($\leqslant K$)-levels, with applications, Int. J. Comput. Geom. Appl. 60 (1996) 247–261.
[12] S. Fekete, On the complexity of min-link red-blue separation, Manuscript, 1992.
[13] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, Inf. Process. Lett. 33 (1989) 169–174.
[14] M.F. Houle, Weak Separability of Sets, PhD thesis, McGill University, 1989.
[15] M.F. Houle, Algorithms for weak and wide separation of sets, Discrete Appl. Math. 45 (1993) 139–159.
[16] F. Hurtado, M. Noy, P.A. Ramos, C. Seara, Separating objects in the plane by wedges and strips, Discrete Appl. Math. 109 (2001) 109–138.
[17] M. van Kreveld, T. van Lankveld, R. Veltkamp, Identifying well-covered minimal bounding rectangles in 2D point data, in: Abstracts 25th Europ. Workshop Comput. Geom., 2009, pp. 277–280.
[18] M. Löffler, Data Imprecision in Computational Geometry, PhD thesis, Utrecht University, 2009.
[19] M. Löffler, M. van Kreveld, Largest and smallest convex hulls for imprecise points, Algorithmica 56 (2010) 235–269.
[20] M. Löffler, M. van Kreveld, Largest bounding box, smallest diameter, and related problems on imprecise points, Comput. Geom. Theory Appl. 43 (2010) 419–433.
[21] J. Matoušek, Geometric Discrepancy, Springer, Berlin, Heidelberg, 2010.
[22] J. Matoušek, Approximation and optimal geometric divide-and-conquer, J. Comput. Syst. Sci. 50 (2) (1995) 203–208.
[23] N. Megiddo, Linear-time algorithms for linear programming in $\mathbb{R}^3$ and related problems, SIAM J. Comput. 12 (1983) 759–776.
[24] J.S.B. Mitchell, Approximation Algorithms for Geometric Separation Problems, Technical Report, State University of New York at Stony Brook, 1993.
[25] Y. Myers, L. Joskowicz, Uncertain geometry with dependencies, in: Proc. 14th ACM Symp. Solid Phys. Mod, 2010, pp. 159–164.
[26] T. Ottmann, E. Soisalon-Soininen, D. Wood, On the definition and computation of rectilinear convex hulls, Inf. Sci. 33 (3) (1984) 157–171.
[27] J. O'Rourke, S. Rao Kosaraju, N. Megiddo, Computing circular separability, Discrete Comput. Geom. 1 (1986) 105–113.
[28] D. Salesin, J. Stolfi, L.J. Guibas, Epsilon geometry: building robust algorithms from imprecise computations, in: Proc. 5th ACM Symp. Comput. Geom, 1989, pp. 208–217.
[29] C. Seara, On Geometric Separability, PhD thesis, Universidad Politécnica de Catalunya, 2002.
[30] F. Sheikhi, A. Mohades, M. de Berg, M. Davoodi, Separating bichromatic point sets by L-shapes, Comput. Geom. Theory Appl. 48 (9) (2015) 673–687.
[31] S. Suri, K. Verbeek, H. Yildiz, On the most likely convex hull of uncertain points, in: Proc. 21st Europ. Symp. Alg, 2013, pp. 791–802.