Robust Bichromatic Classification using Two Lines

Erwin Glazenburg[1], Thijs van der Horst[1,2], Tom Peters[2], Bettina Speckmann[2], and Frank Staals[1]

1 **Utrecht University**
`[e.p.glazenburg, t.w.j.vanderhorst, f.staals]@uu.nl`

2 **TU Eindhoven**
`[t.peters1, b.speckmann]@tue.nl`

Abstract

Given two sets R and B of at most n points in the plane, we present efficient algorithms to find a two-line linear classifier that best separates the "red" points in R from the "blue" points in B and is robust to outliers. More precisely, we find a region $\mathcal{W}_B$ bounded by two lines, so either a halfplane, strip, wedge, or double wedge, containing (most of) the blue points B, and few red points. Our running times vary between optimal $O(n \log n)$ and $O(n^4)$, depending on the type of region $\mathcal{W}_B$ and whether we wish to minimize only red outliers, only blue outliers, or both.

Related Version A full version can be found on arXiv [5]

1 Introduction

Let R and B be two sets of at most n points in the plane. Our goal is to best separate the "red" points R from the "blue" points B using at most two lines. That is, we wish to find a region $\mathcal{W}_B$ bounded by lines ℓ_1 and ℓ_2 containing (most of) the blue points B, so that the number k_R of points from R in the interior $\text{int}(\mathcal{W}_B)$ of $\mathcal{W}_B$ and/or the number k_B of points from B in the interior $\text{int}(\mathcal{W}_R)$ of the region $\mathcal{W}_R = \mathbb{R}^2 \setminus \mathcal{W}_B$ is minimized. We refer to these sets of red and blue outliers as $\mathcal{E}_R = R \cap \text{int}(\mathcal{W}_B)$ and $\mathcal{E}_B = B \cap \text{int}(\mathcal{W}_R)$, respectively, and define $\mathcal{E} = \mathcal{E}_R \cup \mathcal{E}_B$ and $k = k_R + k_B$.

Region $\mathcal{W}_B$ is either: (i) a halfplane, (ii) a *strip* bounded by two parallel lines ℓ_1 and ℓ_2, (iii) a *wedge*, i.e. one of the four regions induced by a pair of intersecting lines ℓ_1, ℓ_2, or (iv) a *double wedge*, i.e. two opposing regions induced by a pair of intersecting lines ℓ_1, ℓ_2. See Figure 1. We can reduce the case that $\mathcal{W}_B$ would consist of three regions to the single-wedge case, by recoloring the points. For each of these cases for the shape of $\mathcal{W}_B$ we consider three problems: allowing only red outliers ($k_B = 0$) and minimizing k_R, allowing only blue outliers ($k_R = 0$) and minimizing k_B, or allowing both outliers and minimizing k. We present efficient algorithms for each of these problems, see Table 1.

Related work. Binary classification is a key problem in computer science. Linear classifiers such as SVMs [3] compute a hyperplane separating R and B; when R and B are not linearly

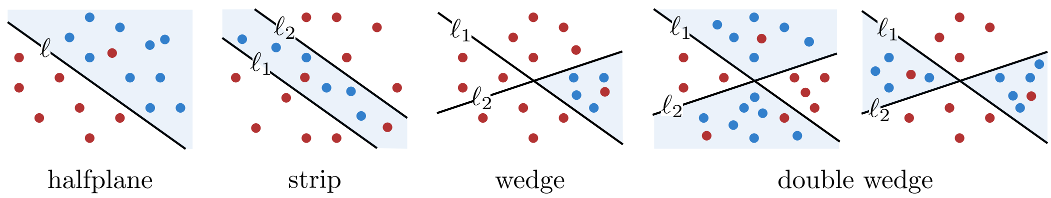

Figure 1 We consider separating R and B by at most two lines. This gives rise to four types of regions $\mathcal{W}_B$; halfplanes, strips, wedges, and two types of double wedges; hourglasses and bowties.

40th European Workshop on Computational Geometry, Ioannina, Greece, March 13–15, 2024.
This is an extended abstract of a presentation given at EuroCG'24. It has been made public for the benefit of the community and should be considered a preprint rather than a formally reviewed paper. Thus, this work is expected to appear eventually in more final form at a conference with formal proceedings and/or in a journal.

region $\mathcal{W}_B$	minimize k_R	minimize k_B	minimize k
halfplane	$O(n \log n)$ $\star$	$O(n \log n)$ $\star$	$O((n + k^2) \log n)$ [2]
strip	$\Theta(n \log n)$ [8], §3	$O(n^2 \log n)$ $\star$	$O(n^2 \log n)$ $\star$
wedge	$O(n^2)$ [8]	$O(n^2 k_B$	$O((n^2 k + nk^3)$
	$O(n \log n)$ §4	$\log n \log k_B)$ $\star$	$\log n \log k)$ $\star$
double (bowtie) wedge	$O(n^2)$ §5	$O(n^2 \log n)$ $\star$	$O(n^4)$ $\star$

Table 1 An overview of our results. A star $\star$ means this result is shown in the full version.

separable like in Figure 2 one could try using different (non-linear) separators, or allowing for outliers. Hurtato et al. [6, 7] give $O(n \log n)$ algorithms for perfectly separating R and B using two lines (i.e. a strip, wedge or double wedge) without outliers, which are optimal [1]. Alternatively, Chan [2] presented algorithms for linear programming in constant dimension that allow for up to k violations, and thus solve hyperplane separation with up to k outliers.

A combination of the above, i.e. using more general separators while giving guarantees on the number of outliers, seems to be less well studied. Seara [8] showed how to compute a strip containing all blue points and minimal red points in $O(n \log n)$ time, and a wedge with the same properties in $O(n^2)$ time. In this paper, we take some further steps toward the fundamental problem of computing robust non-linear separators with performance guarantees.

Results. We present efficient algorithms for computing a region $\mathcal{W}_B$ (strip, wedge, or double wedge) that minimizes red (k_R), blue (k_B), or both (k) outliers. Refer to Table 1 for an overview. In this extended abstract we focus on three entries of Table 1: minimizing k_R for strips (Section 3), wedges (Section 4), and double wedges (Section 5). The other results and omitted proofs can be found in the full version [5] on arXiv.

Most notably, our optimal $\Theta(n \log n)$ algorithm for computing a wedge minimizing k_R improves the earlier $O(n^2)$ time algorithm from Seara [8]. We also provide the first algorithms for minimizing k_B for strips, wedges, and double wedges, and surprisingly these problems seem more difficult than their counterpart of minimizing k_R.

2 Preliminaries

We assume $B \cup R$ contains at least three points and is in general position, i.e. no two points have the same x- or y-coordinate, and no three points are co-linear.

Notation. Let ℓ^- and ℓ^+ be the two halfplanes bounded by line ℓ, with ℓ^- below ℓ (or left of ℓ if ℓ is vertical). Any pair of lines ℓ_1 and ℓ_2, with the slope of ℓ_1 smaller than that of ℓ_2, subdivides the plane into at most four interior-disjoint regions North$(\ell_1, \ell_2) = \ell_1^+ \cap \ell_2^+$, East$(\ell_1, \ell_2) = \ell_1^+ \cap \ell_2^-$, South$(\ell_1, \ell_2) = \ell_1^- \cap \ell_2^-$ and West$(\ell_1, \ell_2) = \ell_1^- \cap \ell_2^+$. When ℓ_1 and ℓ_2

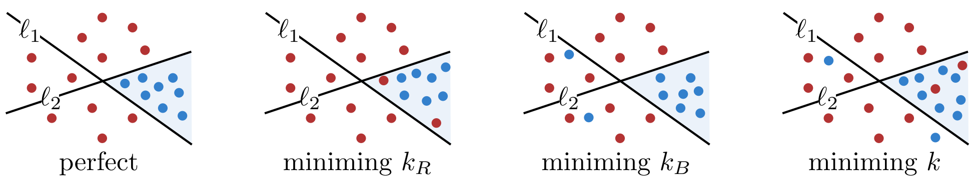

Figure 2 When considering outliers, we may allow only red outliers, only blue outliers, or both.

are clear from the context we may simply write North to mean North(ℓ_1, ℓ_2) etc. We assign each of these regions to either B or R, so that $\mathcal{W}_B$ $(= \mathcal{W}_B(\ell_1, \ell_2))$ and $\mathcal{W}_R$ $(= \mathcal{W}_R(\ell_1, \ell_2))$ are the union of some elements of $\{\text{North}, \text{East}, \text{South}, \text{West}\}$. In case ℓ_1 and ℓ_2 are parallel, we assume that ℓ_1 lies below ℓ_2, and thus $\mathcal{W}_B = \text{East}$.

Duality. We make frequent use of the standard point-line duality [4], where we map objects in *primal* space to objects in a *dual* space. In particular, a primal point $p = (a, b)$ is mapped to the dual line $p^* : y = ax - b$ and a primal line $\ell : y = ax + b$ is mapped to the dual point $\ell^* = (a, -b)$. If primal point p lies above line ℓ, then dual line p^* lies below point ℓ^*.

For a set of lines L, we are often interested in the *arrangement* $\mathcal{A}(L)$, i.e. the vertices, edges, and faces formed by the lines in L. Let $\mathcal{U}(L)$ be the upper envelope of L, i.e. the polygonal chain following the highest line in $\mathcal{A}(L)$, and $\mathcal{L}(L)$ the lower envelope.

Property of an optimal wedge. It can be shown that, for any (double) wedge classification problem, there exists an optimum where both lines go through a blue and a red point. Therefore there exists a somewhat simple $O(n^4)$ algorithm for finding (double) wedges minimizing either k_R, k_B, or k, which considers all pairs of lines through red and blue points.

3 Strip separation with red outliers

We first consider the case where W_B forms a strip, bounded by parallel lines ℓ_1 and ℓ_2, with ℓ_2 above ℓ_1. We want B to be inside the strip, and R outside, and here we show how to minimize red outliers k_R. We do this in the dual, where we want to find two points ℓ_1^* and ℓ_2^* with the same x-coordinate such that vertical segment $\overline{\ell_1^* \ell_2^*}$ intersects all blue lines and as few red lines as possible. Note that ℓ_1^* must be above $\mathcal{U}(B^*)$ and ℓ_2^* must be below $\mathcal{L}(B^*)$. Since shortening a segment can not make it intersect more red lines, we can even assume they lie exactly on the envelopes.

As $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ are x-monotone, there is only one degree of freedom for choosing our segment: its x-coordinate. We parameterize $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ over $\mathbb{R}$, our one-dimensional *parameter space*, such that each point $p \in \mathbb{R}$ corresponds to the vertical segment $\overline{\ell_1^* \ell_2^*}$ on the line $x = p$. We wish to find a point in this parameter space, i.e. an x-coordinate, whose corresponding segment minimizes the number of red misclassifications. Let the *forbidden regions* of a red line r be those intervals on the parameter space in which corresponding segments intersect r. We distinguish between four types of red lines, as in Figure 3:

- Line a intersects $\mathcal{U}(B^*)$ in points a_1 and a_2, with $a_1 \leq a_2$. Segments with ℓ_1^* left of a_1 or right of a_2 misclassify a, so a produces two forbidden intervals: $(-\infty, a_1)$ and (a_2, ∞).
- Line b intersects $\mathcal{L}(B^*)$ in points b_1 and b_2, with $b_1 \leq b_2$. Similar to line a this produces forbidden intervals $(-\infty, b_1)$ and (b_2, ∞).
- Line c intersects $\mathcal{L}(B^*)$ in c_1 and $\mathcal{U}(B^*)$ in c_2. Only segments between c_1 and c_2 misclassify c. This gives one forbidden interval: $(\min\{c_1, c2\}, \max\{c_1, c_2\})$.
- Line d intersects neither $\mathcal{U}(B^*)$ nor $\mathcal{L}(B^*)$. All segments misclassify d. This gives one trivial forbidden region, namely the entire space $\mathbb{R}$.

The above list is exhaustive. To see this, note that the two lines supporting the unbounded edges of $\mathcal{U}(B^*)$ also support the unbounded edges of $\mathcal{L}(B^*)$.

Our goal is to find a point that lies in as few of these forbidden regions as possible. We can compute such a point in $O(n \log n)$ time by sorting and scanning. Computing $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ takes $O(n \log n)$ time. Given a red line $r \in R^*$ we can compute its intersection points

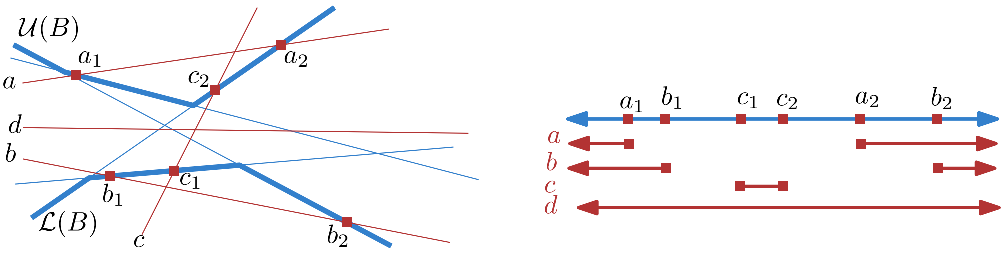

Figure 3 Four types of red lines for strip separation, with restrictions on their parameter space.

with $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ in $O(\log n)$ time using binary search (since $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ are convex). Computing the forbidden regions thus takes $O(n \log n)$ time in total. We conclude:

▶ **Theorem 3.1.** *Given two sets of n points $B, R \subset \mathbb{R}^2$, we can construct a strip $\mathcal{W}_B$ minimizing the number of red outliers k_R in $O(n \log n)$ time.*

4 Wedge separation with red outliers

We consider the case where the region $\mathcal{W}_B$ is a single wedge and $\mathcal{W}_R$ is the other three wedges. Here we show how to compute an optimal East or West wedge minimizing red outliers, i.e. we compute two lines ℓ_1 and ℓ_2 such that every blue point and as few red points as possible lie above ℓ_1 and below ℓ_2. In the dual this corresponds to two points ℓ_1^* and ℓ_2^* such that all blue lines and as few red lines as possible lie below ℓ_1^* and above ℓ_2^*, as in Figure 4. In the full version, we compute an optimal North or South wedge in a similar way.

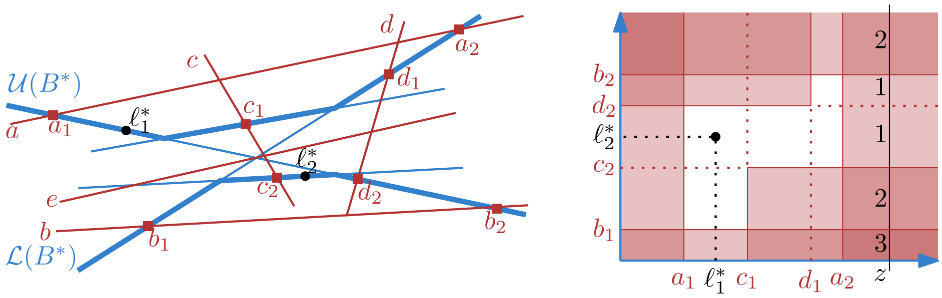

Figure 4 The arrangement of $B^* \cup R^*$ with its parameter space and forbidden regions.

Clearly ℓ_1^* must lie above $\mathcal{U}(B^*)$, and ℓ_2^* below $\mathcal{L}(B^*)$; as in the strip case, we can even assume they lie exactly on $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$. Similar to the case of strips we parameterize $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$ over $\mathbb{R}^2$, such that a point (p, q) in this two-dimensional parameter space corresponds to two dual points ℓ_1^* and ℓ_2^*, with ℓ_1^* on $\mathcal{U}(B^*)$ at $x = p$ and ℓ_2^* on $\mathcal{L}(B^*)$ at $x = q$. See Figure 4. We wish to find a value in our parameter space whose corresponding segment minimizes the number of red misclassifications. Let the forbidden regions of a red line r again be those regions in the parameter space in which corresponding segments misclassify r. We distinguish between five types of red lines, as in Figure 4 (left):

- Line a intersects $\mathcal{U}(B^*)$ in a_1 and a_2, with a_1 left of a_2. Only segments with ℓ_1^* left of a_1 or right of a_2 misclassify a. This produces two forbidden regions: $(-\infty, a_1) \times (-\infty, \infty)$ and $(a_2, \infty) \times (-\infty, \infty)$.
- Line b intersects $\mathcal{L}(B^*)$ in b_1 and b_2, with b_1 left of b_2. Symmetric to line a this produces forbidden regions $(-\infty, \infty) \times (-\infty, b_1)$ and $(-\infty, \infty) \times (b_2, \infty)$.

- Line c intersects $\mathcal{U}(B^*)$ in c_1 and $\mathcal{L}(B^*)$ in c_2, with c_1 left of c_2. Only segments with endpoints after c_1 and before c_2 misclassify c, producing the region $(c_1, \infty) \times (-\infty, c_2)$. (Segments with endpoints before c_1 and after c_2 do intersect c, but do not misclassify it)
- Line d intersects $\mathcal{U}(B^*)$ in d_1 and $\mathcal{L}(B^*)$ in d_2, with d_1 right of d_2. Symmetric to line c it produces the forbidden region $(-\infty, d_1) \times (d_2, \infty)$.
- Line e intersects neither $\mathcal{U}(B^*)$ nor $\mathcal{L}(B^*)$. All segments misclassify e. This produces one forbidden region; the entire plane $\mathbb{R}^2$.

Our goal is again to find a point that lies in as few of these forbidden regions as possible. Since all regions are axis-aligned rectangles, we can do so using a simple sweepline algorithm in $O(n \log n)$ time. Constructing $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$, finding the intersections of every red line r with $\mathcal{U}(B^*)$ and $\mathcal{L}(B^*)$, determining the type of r $(a - e)$, and constructing its forbidden regions all take $O(n \log n)$ time as well.

▶ **Theorem 4.1.** *Given two sets of n points $B, R \subset \mathbb{R}^2$, we can construct an* East *or* West *wedge containing all points of B and the fewest points of R in $O(n \log n)$ time.*

5 Double wedge separation with red outliers

Although the wedge algorithm was a direct extension of the strip algorithm, the double wedge algorithm uses different techniques, which we briefly review; see the full version for details. We consider finding a *bowtie* wedge $\mathcal{W}_B$ while minimizing red outliers, i.e. all of B and as little of R as possible lies in the West and East wedge. In the dual this corresponds to a line segment intersecting all of B^*, and as little of R^* as possible.

Observe that a segment intersecting all lines of B^* must have endpoints in antipodal outer faces of $\mathcal{A}(B^*)$, i.e. two opposite outer faces sharing the same two infinite bounding lines. For all $O(n)$ pairs of antipodal faces, we could apply a very similar algorithm to the wedge algorithm in Section 4, resulting in $O(n \cdot n \log n) = O(n^2 \log n)$ time.

Alternatively, we construct the entire arrangement $\mathcal{A}(B^* \cup R^*)$ of all lines explicitly in $O(n^2)$ time (see e.g. [4]). Consider a pair of faces P and Q that are antipodal in $\mathcal{A}(B^*)$, and assume w.l.o.g. they are separated by the x-axis, with P above Q. There are two types of red lines: *splitting* lines that intersect both P and Q once, and *stabbing* lines that intersect at most one of P and Q, see Figure 5. A red line is a splitting line for exactly one pair of antipodal faces, while it can be a stabbing line for multiple pairs. Recall that we wish to find a segment from P to Q intersecting as few red lines as possible. The s splitting lines divide the boundary of P and Q into $s + 1$ chains $P_0..P_s$ $(Q_0..Q_s)$. Within one such chain P_i on P we only need to consider the point p_i with the most stabbing lines above it: a segment from p_i to Q will not intersect those lines, since Q is below P_i. Similarly, we only need to consider point q_j on chain Q_j with the most stabbing lines below it. Using dynamic programming we can then find the pair of chains P_i, Q_j such that $\overline{p_i q_j}$ intersects the fewest red lines in $O(n + s^2)$ time. Doing so for all pairs of antipodal faces yields a total running time of $O(n^2)$.

▶ **Theorem 5.1.** *Given two sets of n points $B, R \subset \mathbb{R}^2$, we can construct the bowtie double wedge $\mathcal{W}_B$ minimizing the number of red outliers k_R in $O(n^2)$ time.*

Consider the related problem of finding a bowtie wedge while minimizing k_B, which we solve in $O(n^2 \log n)$ time in the full version. Note that we can not just recolor the points and use the above $O(n^2)$ time algorithm: after recoloring, we would wish to find a blue hourglass wedge minimizing k_R, which is a different problem. Therefore, unfortunately, finding any double wedge (bowtie or hourglass) while minimizing k_R still takes $O(n^2 \log n)$ time.

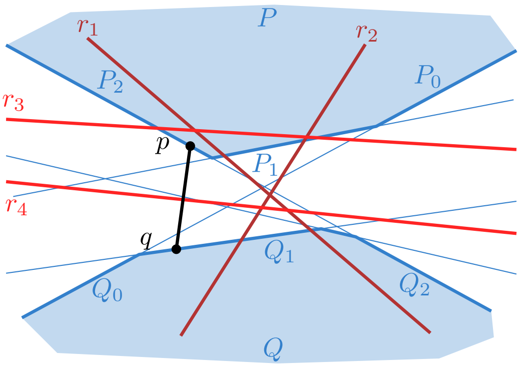

Figure 5 Two antipodal faces P and Q, with two splitting lines r_1, r_2 and two stabbing lines r_3, r_4, and an optimal segment $\overline{pq}$ from P to Q.

References

1 Esther M. Arkin, Ferran Hurtado, Joseph S. B. Mitchell, Carlos Seara, and Steven Skiena. Some lower bounds on geometric separability problems. *International Journal of Computational Geometry & Applications*, 16(1):1–26, 2006. doi:10.1142/S0218195906001902.

2 Timothy M. Chan. Low-dimensional linear programming with violations. *SIAM Journal on Computing*, 34(4):879–893, 2005. doi:10.1137/S0097539703439404.

3 Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, Sep 1995. doi:10.1007/BF00994018.

4 Mark de Berg, Otfried Cheong, Marc J. van Kreveld, and Mark H. Overmars. *Computational geometry: algorithms and applications, 3rd Edition*. Springer, 2008.

5 Erwin Glazenburg, Thijs van der Horst, Tom Peters, Bettina Speckmann, and Frank Staals. Robust bichromatic classification using two lines, 2024. arXiv:2401.02897.

6 Ferran Hurtado, Mercè Mora, Pedro A. Ramos, and Carlos Seara. Separability by two lines and by nearly straight polygonal chains. *Discrete Applied Mathematics*, 144(1-2):110–122, 2004. doi:10.1016/j.dam.2003.11.014.

7 Ferran Hurtado, Marc Noy, Pedro A. Ramos, and Carlos Seara. Separating objects in the plane by wedges and strips. *Discrete Applied Mathematics*, 109(1-2):109–138, 2001. doi:10.1016/S0166-218X(00)00230-4.

8 Carlos Seara. *On geometric separability*. PhD thesis, Univ. Politecnica de Catalunya, 2002.