Finding enclosing boxes with empty intersection

C. Cortés*, J.M. Díaz-Báñez† y J. Urrutia ‡

Abstract

Let S be a point set in general position on the plane such that its elements are colored red or blue. We study the following problem: Remove as few points as possible from S such that the remaining points can be enclosed by two isothetic rectangles, one containing all the red points, the other all the blue points, and such that each rectangle contains only points of one color. We prove that this problem can be solved in $O(n^3)$ time and space.

1 Introduction

In Data Mining and Classification problems, a natural method to analyse data, is to select prototypes representing different classes of data. A standard technique to achieve this, is to perform cluster analysis on the training data [4, 6]. The clustering can be obtained by using simple geometric shapes such as circles or boxes. In [1, 5], circles and parallel-axis boxes respectively, are considered for the selection. In [1], the following problem is studied: given a bicolored point set, find a ball that contains the maximum number of red points without containing any blue point inside it.

In some cases these methods can produce slanted classifications due to the fact that some data may be defective or contain values out of reasonable ranges. In other cases, we may obtain data hard to classify due to relatively small similarities between different classes. A possible way to find a better classification for the former problem is to remove some data-points from the input. Culling the minimum number of such points can be a suitable criterium to lose as less information as possible. Thus, in this paper we study the following problem: Let S be a bicolored point set in general position on the plane such that no two elements of S lie on a vertical or horizontal line. Find the largest subset S' of S that can be enclosed by two isothetic rectangles $\mathcal{R}$ and $\mathcal{B}$ such that:

- $\mathcal{R}$ (resp. $\mathcal{B}$) contains all the red (resp. blue) points of S' respectively
- $\mathcal{R}$ (resp. $\mathcal{B}$) contains no blue (resp. red) points of S'.

We will refer to this problem as *Empty Intersection Enclosing Boxes* problem or simply as *EIEB-problem*. For example, the solution to the *EIEB-problem* for the point set shown in Figure 1 is 2, since by removing the points r_1 and b_1 we can obtain two rectangles, $\mathcal{R}$ and $\mathcal{B}$ each of them containing only red and blue points respectively.

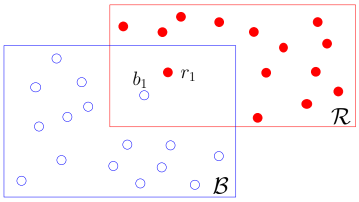

Figure 1: Removing points r_1 and b_1, we get a solution.

From now on, an isothetic rectangle enclosing a set of red (resp., blue) points will be called *red rectangle*, denoted by $\mathcal{R}$ (resp., *blue rectangle*, denoted by $\mathcal{B}$).

To solve our problem, we observe first that given a bicolored point set S, and two rectangles $\mathcal{R}$ and $\mathcal{B}$ that provide an optimal solution to the *EIEB-problem* for S, there are three types of relative positions of $\mathcal{R}$ with respect to $\mathcal{B}$, up to symmetry. These are depicted in Figure 2. We call a *corner solution* to that in which $\mathcal{R}$ contains exactly one corner of $\mathcal{B}$; a *sandwich solution* to that in which $\mathcal{R}$ intersects properly two parallel sides of $\mathcal{B}$; and *disjoint solution* to that in which $\mathcal{R}$ and $\mathcal{B}$ can be separated either by a horizontal or a vertical line.

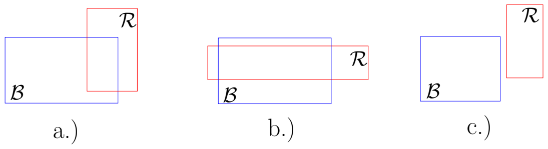

Figure 2: a.) Corner, b.) sandwich and, c.) disjoint solutions.

Our procedure consists on looking for the best solution of each type keeping the best among them. In all of the previous cases, we reduce our 2-dimensional problem to the following 1-dimensional problem:

Maximum Consecutive Subsequence (MCS): Given a sequence $x_1, x_2, ..., x_n$ of $0's$, $+1$'s and $-1's$,

*Departamento Matemática Aplicada I, Universidad de Sevilla, ccortes@us.es

†Departamento Matemática Aplicada II, Universidad de Sevilla, dbanez@us.es

‡Instituto de Matemáticas, Universidad Nacional Autónoma de México, urrutia@matem.unam.mx

compute, for every index $i = 1, ..., n$, all the subsequences $x_i, x_{i+1}, ..., x_j$ of consecutive elements such that $x_i + x_{i+1} + \ldots + x_j$ is maximized over all subsequences starting at x_i.

It is relatively easy to see that the *MCS-problem* can be solved in linear time by using the same techniques used to solve Bentley's *Maximum segment sum* problem [2].

2 Finding the optimum Corner Solution

We now sketch the key idea to find the best *corner solution*. The remaining cases are solved using similar techniques.

Let $(\mathcal{R}, \mathcal{B})$ be a corner type pair of rectangles. Assume for the rest of this section that, as in Figure 3, $\mathcal{R}$ contains the topmost right corner of $\mathcal{B}$. We denote by $Red(\mathcal{R} \setminus \mathcal{B})$ to the set of red points of S contained in $\mathcal{R} \setminus \mathcal{B}$. Similarly we define $Blue(\mathcal{R} \setminus \mathcal{B})$, $Red(\mathcal{B} \setminus \mathcal{R})$, $Blue(\mathcal{B} \setminus \mathcal{R})$, $Red(\mathcal{R} \cap \mathcal{B})$, and $Blue(\mathcal{R} \cap \mathcal{B})$, see Figure 3. We remark that $\mathcal{R}$ and $\mathcal{B}$ are considered to be closed sets.

Proposition 1 *Let $(\mathcal{R}, \mathcal{B})$ be a corner type pair of rectangles. Then, it is possible to find another corner type pair $(\hat{\mathcal{R}}, \hat{\mathcal{B}})$ such that $\hat{\mathcal{R}} \setminus \hat{\mathcal{B}}$ (resp., $\hat{\mathcal{B}} \setminus \hat{\mathcal{R}}$) contains at least $Red(\mathcal{R} \setminus \mathcal{B})$ red points (resp., $Blue(\mathcal{B} \setminus \mathcal{R})$ blue points) and the sides of $\hat{\mathcal{R}}$ (resp., $\hat{\mathcal{B}}$) go through red (resp., blue) points.*

Corollary 2 *There exists a pair $(\mathcal{R}, \mathcal{B})$ of corner type rectangles that provides an optimal* corner solution *such that the sides of $\mathcal{R}$ (resp., $\mathcal{B}$) go through red (resp., blue) points.*

From now on, any rectangle $\mathcal{R}$ (resp., $\mathcal{B}$) will be considered to be delimited by red (resp., blue) points of S.

A pair $(\mathcal{R}, \mathcal{B})$ of corner type rectangles that provides an optimal *corner solution* will be a pair that maximizes the sum $|Red(\mathcal{R} \setminus \mathcal{B})| + |Blue(\mathcal{B} \setminus \mathcal{R})|$.

Let $\mathcal{Q}_\mathcal{R}$ be the quadrant obtained from $\mathcal{R}$ by extending to infinity its left and lower sides towards the North and the East respectively. We will refer to $\mathcal{Q}_\mathcal{R}$ as the *red quadrant*. Similarly we define the *blue quadrant* $\mathcal{Q}_\mathcal{B}$, obtained by extending to infinity the right and upper sides of $\mathcal{B}$ towards the South and West respectively. We will assume that the quadrants include their borders.

As a consequence of Proposition 1, if $(\mathcal{R}, \mathcal{B})$ provides an optimal solution, then $|Red(\mathcal{R} \setminus \mathcal{B})| = |Red(\mathcal{Q}_\mathcal{R} \setminus \mathcal{Q}_\mathcal{B})|$ and $|Blue(\mathcal{R} \setminus \mathcal{B})| = |Blue(\mathcal{Q}_\mathcal{B} \setminus \mathcal{Q}_\mathcal{R})|$. We then reformulate our problem as follows: find the pair of quadrants $(\mathcal{Q}_\mathcal{R}, \mathcal{Q}_\mathcal{B})$ that maximize the sum $|Red(\mathcal{Q}_\mathcal{R} \setminus \mathcal{Q}_\mathcal{B})| + |Blue(\mathcal{Q}_\mathcal{B} \setminus \mathcal{Q}_\mathcal{R})|$.

It is easy to see that by using *range search* techniques [3], we can solve this problem in $O(n^4)$ time.

We proceed now to show how to solve the *EIEB-problem* in $O(n^3)$ time.

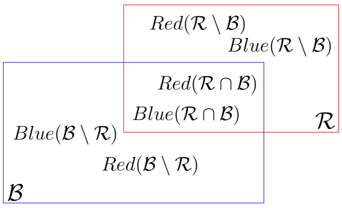
$Red(\mathcal{R} \setminus \mathcal{B})$
$Blue(\mathcal{R} \setminus \mathcal{B})$
$Red(\mathcal{R} \cap \mathcal{B})$
$Blue(\mathcal{R} \cap \mathcal{B})$
$\mathcal{R}$
$Blue(\mathcal{B} \setminus \mathcal{R})$
$Red(\mathcal{B} \setminus \mathcal{R})$
$\mathcal{B}$

Figure 3: Notation for points in S depending on their location.

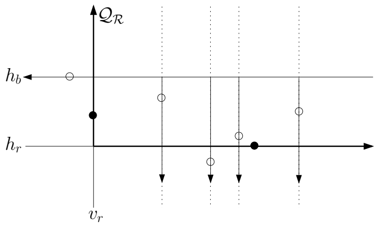

Figure 4: Illustrating the Query problem.

2.1 The Algorithm

Consider the orthogonal grid generated by drawing horizontal and vertical lines through the elements of S.

Assume we have already preprocessed points in S so that it is possible to answer in $O(1)$ *amortized time* the next problem (Figure 4):

OptQuad: Given a red quadrant $\mathcal{Q}_\mathcal{R}$, determined by two red lines $< h_r, v_r >$, and a blue horizontal line h_b above h_r, find the blue vertical line v_b to the right of v_r such the blue quadrant $\mathcal{Q}_\mathcal{B}$ bounded above and to its right by h_b and v_r respectively, is such that it maximizes the sum $|Red(\mathcal{Q}_\mathcal{R} \setminus \mathcal{Q}_\mathcal{B})| + |Blue(\mathcal{Q}_\mathcal{B} \setminus \mathcal{Q}_\mathcal{R})|$ over all the possible choices of v_b.

Our algorithm to solve an instance of a *corner type* solution proceeds as follows.

***Corner Solution* Algorithm (CS-Algorithm):**

1. For each vertex of the grid, compute the number of red and blue points of S laying in the four (North-West, North-East, South-West and South East) quadrants with vertex in it.

2. For each red quadrant $\mathcal{Q}_\mathcal{R} < h_r, v_r >$ and for every blue horizontal line h_b above h_r, the query reports a blue vertical line v_b. Store the sum $s_{(\mathcal{Q}_\mathcal{R}, \mathcal{Q}_\mathcal{B})} = |Red(\mathcal{Q}_\mathcal{R} \setminus \mathcal{Q}_\mathcal{B})| + |Blue(\mathcal{Q}_\mathcal{B} \setminus \mathcal{Q}_\mathcal{R})|$.

3. Output the pair $(\mathcal{Q}_\mathcal{R}, \mathcal{Q}_\mathcal{B})$ that provides the maximum value $s_{(\mathcal{Q}_\mathcal{R}, \mathcal{Q}_\mathcal{B})}$.

Complexity: It is easy to see that the first step of our algorithm can be completed in quadratic time [3]. The second step of our algorithm answers $O(n^3)$ queries. These queries can be solved in amortized constant time using **OptQuad**. Finally reporting the best $s_{(\mathcal{Q}_\mathcal{R},\mathcal{Q}_\mathcal{B})}$ can be done in constant time.

2.2 The Preprocessing

We now describe briefly the preprocessing needed to solve **OptQuad** and prove the correctness of the whole algorithm. Consider the orthogonal grid obtained by passing a horizontal and a vertical line through every element of S. Assume that these lines are colored red or blue according to the color of the point in S they contain.

Each pair consisting of a horizontal blue line h_b and a horizontal red line h_r below it determines a horizontal strip HS_{h_b,h_r}. We assign weights to some elements of S according to the following criteria, see Figure 5 a.):

1. Every red point inside HS_{h_b,h_r} has weight -1
2. Red points in or above HS_{h_b,h_r} have weight 0
3. Blue points in HS_{h_b,h_r} have weight 0
4. Blue points below HS_{h_b,h_r} have weight $+1$

Blue points above HS_{h_b,h_r} and red points below HS_{h_b,h_r} are discarded. We next project our blue and red points together with their weights on the $x-axis$ obtaining a sequence $\mathcal{P} = \{p_{\sigma(1)}, \dots, p_{\sigma(k)}\}$ of points with weights $-1, 0$, or $+1$, where k is the number of weighted points of S. We use now the solution to the *MCS* problem, and find for each $p_{\sigma(i)}$ the $j > i$ s.t. the sum of the weights of all the elements in $\mathcal{P}$ between $p_{\sigma(i)}$ and $p_{\sigma(j)}$ (including the weight of $p_{\sigma(j)}$) is maximized.

Suppose now that we have a red quadrant bounded below by h_r and to its left by a vertical line v_r through a red point above h_r, and a blue quadrant bounded above by h_b and to its right by a vertical line v_b through a blue point as shown in Figure 5 b.). Let $\mathcal{Q}'$ be the quadrant bounded above by h_b, and to the right by v_r.

Let us define the following numbers:

- Let b_1 be the number of blue points in $\mathcal{Q}'$
- Let r_1 be the number of red points in $\mathcal{Q}_\mathcal{R}$
- Let c be the sum of the weights of the elements of $\mathcal{P} = \{p_{\sigma(1)}, \dots, p_{\sigma(k)}\}$ between $p_{\sigma(i)}$ and $p_{\sigma(j)}$, where $p_{\sigma(i)}$ and $p_{\sigma(j)}$ are the points into which points in v_r and v_b were projected in $\mathcal{P} = \{p_{\sigma(1)}, \dots, p_{\sigma(k)}\}$.

Theorem 3 *The number of red points in $\mathcal{Q}_\mathcal{R}$ plus the number of blue points in $\mathcal{Q}_\mathcal{B}$ minus the number of blue and red points in $\mathcal{Q}_\mathcal{R} \cap \mathcal{Q}_\mathcal{R}$ equals $b_1 + r_1 + c$.*

It follows now that we have to maximize c to find the optimal solution in which $\mathcal{Q}_\mathcal{R}$ participates, and $\mathcal{Q}_\mathcal{B}$ is bounded above by h_b. This can be done using the solution to the *SMC problem* in $\mathcal{P} = \{p_{\sigma(1)}, \dots, p_{\sigma(k)}\}$. Thus we have:

Theorem 4 *An optimum corner solution can be found in $O(n^3)$ time and storage, given $O(n^3)$ preprocessing time.*

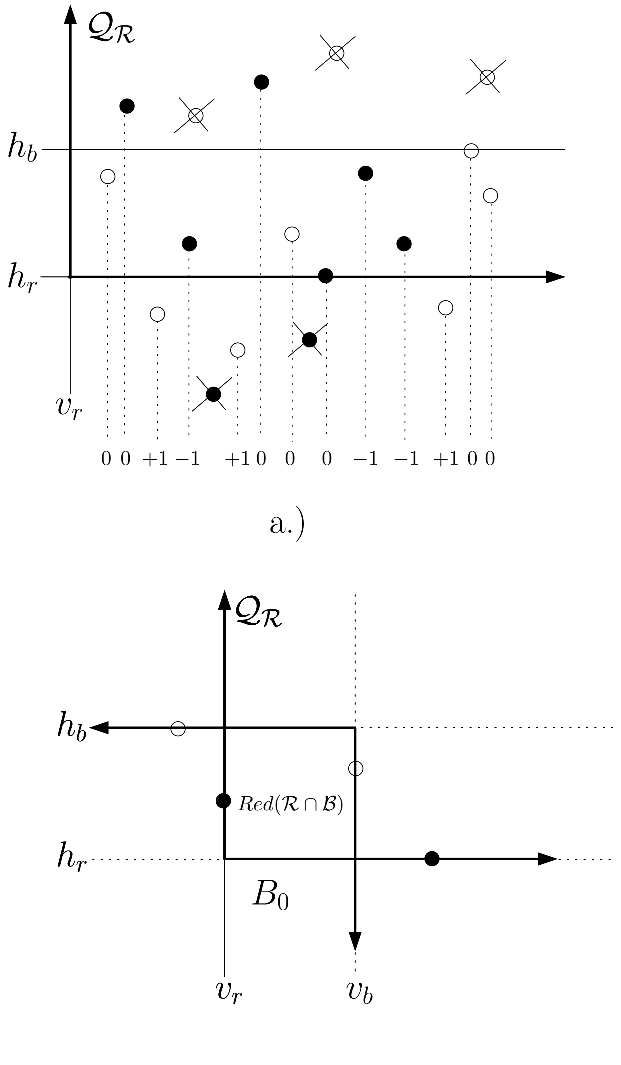
$\mathcal{Q}_\mathcal{R}$
h_b
h_r
v_r
0 0 +1 −1 +1 0 0 0 −1 −1 +1 0 0
a.)
$\mathcal{Q}_\mathcal{R}$
h_b
$Red(\mathcal{R} \cap \mathcal{B})$
h_r
B_0
v_r
v_b
b.)

Figure 5: Illustrating the proof of the algorithm's correctness.

3 Conclusions

We propose an algorithm to solve the *EIEB-problem* that requires $O(n^3)$ time and quadratic space. The algorithm solves separately three different possibilities for an optimal solution type: *corner*, *sandwich* and, *disjoint solution.* We have presented here the way to solve the optimum *corner solution.* This is the most interesting case, the others can be solved in a similar way.

To conclude, let us mention that the 1-dimensional EIEB-problem, where a set of red and blue points on a line are given and we are asked to determine the minimum number of points to be removed in order to get two disjoint intervals containing points of only one color, can be solved in lineal time.

References

[1] B. Aronov, S. Har-Peled. On approximating the depth and related problems. *Proceedings 16th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA 2005), 2005.

[2] J. Bentley, “Programming pearls: algorithm design techniques,” Comm. ACM, vol. 27, no. 9, pp. 865 - 873, 1984.

[3] J.L. Bentley and M.I. Shamos, *A problem in multivariate statistics: Algorithms, data structure and applications*, Proceedings of the 15th annual Allerton Conference on Communications, Control, and Computing, pp. 193-201, 1977.

[4] R. Duda, P. Hart, D. Stork. *Pattern Classification. John Wiley and Sons, Inc.*, New York, 2001.

[5] Eckstein, J., Hammer, P.L., Liu, Y., Nediak,M., Simeone, B. The maximum box problem and its applications to data analysis. Comput. Optim. Appl. 23, 2002, 285-298.

[6] Hand,H., Mannila, H., Smyth, P. Principles of Data Mining. The MIT Press, 2001.