Efficient Certifying Algorithms for Linear Classification

Vincenzo Bonifaci[1][0000−0001−9038−6901]✉ and Sara Galatro[1][0009−0004−6643−727X]

Dipartimento di Matematica e Fisica, Università Roma Tre, Italy
{vincenzo.bonifaci,sara.galatro}@uniroma3.it

Abstract. An efficient certifying algorithm is proposed that, given a set of n points in $\mathbb{R}^d$ with binary labels, either returns a hyperplane separating the points, or identifies $d+2$ of the labeled points that cannot be separated by any hyperplane. The existence of such $d+2$ points in the inseparable case is known to be guaranteed by Kirchberger's theorem in combinatorial geometry; we show how to compute these points efficiently. We then propose a dimension-free and constructive extension of Kirchberger's theorem, where for any $\varepsilon > 0$ one finds either a separating hyperplane, or $O(1/\varepsilon^2)$ of the labeled points that cannot be separated with normalized margin ε by any hyperplane. Our algorithms are based on solving one primal-dual pair of linear programs with d primal and n dual variables, and at most $n-d$ linear equation systems with $O(d)$ equations and $O(d)$ unknowns.

Keywords: Certifying algorithm · Binary classification · Linear duality · Combinatorial convexity · Kirchberger's theorem · Machine learning

1 Introduction

During the last couple of decades, the notion of *certifying algorithm* has been proposed to formalize the idea that algorithms should supplement their answers with certificates, in order to ease the task of checking solution correctness [3, 15, 19]. A certifying algorithm is an algorithm that produces, along with each output, a *certificate* or *witness* that the particular output indeed satisfies the input-output relation required by the computational task at hand. For instance, while a traditional algorithm to check whether a graph is bipartite might only return a yes/no answer, thus requiring its user to blindly trust the computation, a certifying algorithm could instead return a 2-coloring of the graph when the graph is bipartite, and an odd cycle subgraph when the graph is not bipartite; in both cases the end user can easily check the answer's correctness. Additionally, evidence of an odd cycle in the graph makes it simpler for the user to understand *why* the search for a proper bipartition was fruitless.

The notion of certifying algorithm has received attention in several sub-areas of algorithm design, such as graph recognition, and a wealth of significant algorithms have been designed (or redesigned) with the aim of making them certifying; see for example [10, 11, 15, 20, 25] and references therein. In this work, we propose to apply the idea of certifying algorithms to machine learning tasks. Namely, we consider binary linear classification, one of the fundamental tasks in supervised machine learning [22]. In binary linear classification, one is provided with a set of n binary-labeled data points in $\mathbb{R}^d$, and the goal is to construct a hyperplane separating the data points into the two labeled classes (if possible). Clearly, if an algorithm provides a normal vector v to a supposed separating hyperplane, separation can be easily checked by computing inner products between the data points and the vector v. However, if the algorithm fails to construct a separating hyperplane, how can the user easily check that, indeed, no separating hyperplane exists? And can one identify some data points that obstruct the separation?

For binary linear classification, a result in combinatorial geometry known as Kirchberger's theorem ensures that, in the inseparable case, $d+2$ labeled points exist, among the n given, that are inseparable (see Theorem 3 for a formal statement and references). When $d \ll n$, this would provide easily checkable evidence of the impossibility of separation. However, Kirchberger's theorem is existential. In this work, we show how to efficiently *compute* the $d+2$ witness points guaranteed by the theorem. We then provide a constructive, dimension-free extension of Kirchberger's theorem, where for any $\varepsilon > 0$ one finds either a separating hyperplane, or $O(1/\varepsilon^2)$ of the labeled points that cannot be separated with normalized margin ε by any hyperplane (the *normalized margin* being the minimum distance between the data points and the hyperplane, normalized by the diameter of the point set; see Section 4 for a formal definition).

We note that, to some extent, one can see the problem of certifying linear classification as an issue in interpretable machine learning [16, 27]. Citing Lipton et al. [16], "to trust an AI model [...] you might care not only about *how often* a model is right, but also *for which examples* it is right." In this sense, identification

of the witness points guaranteed by Kirchberger's theorem provides the end user of the binary classification method with concrete, combinatorial counter-evidence to the linear separability hypothesis.

Main results and techniques Our main results are twofold:

- We show how Kirchberger's theorem can be made constructive and certifying, by designing an algorithm that given n binary-labeled points in $\mathbb{R}^d$, either returns a hyperplane separating the points, or identifies $d+2$ of the labeled points that cannot be separated by any hyperplane (Theorem 5). The algorithm is based on solving one primal-dual pair of linear programs with d primal and n dual variables, and subsequently solving at most $n-d$ linear systems in $O(d)$ variables and equations each (Algorithm 1).
- We provide a constructive and dimension-free generalization of Kirchberger's theorem, where for any $\varepsilon > 0$ one finds either a separating hyperplane, or $O(1/\varepsilon^2)$ of the labeled points that cannot be separated with normalized margin ε by any hyperplane (Theorem 6). The algorithm is based on linear programming and random sampling (Algorithm 2).

Our techniques are based on linear duality combined with ideas from combinatorial convexity. In particular, we extend proof ideas from *Carathéodory's theorem* [7] and, for the dimension-free generalization, from the so-called *Approximate Carathéodory's theorem* [1, 21, 28]. Both are results in combinatorial convexity: indeed, the inseparability certificates constructed by our algorithms are partly combinatorial in nature. The second algorithm is randomized, as the proof of the Approximate Carathéodory's theorem uses a probabilistic method.

Related work While the notion of efficiently checkable certificates is pervasive in theoretical computer science, viewing certificates as a pragmatic approach to result checking was a later development [5, 19]. According to McConnell et al. [19], the term "certifying algorithm" was first used in [15]. McConnell et al. [19] explicitly define the notion of certifying algorithm and put forward the thesis (that we subscribe to) that "certifying algorithms are much superior to non-certifying algorithms, and that for complex algorithmic tasks, only certifying algorithms are satisfactory." An introduction to certifying algorithms can be found in the survey by Alkassar et al. [3]. Certifying algorithms have been designed for several problems [10, 11, 15, 20, 25], including graph connectivity, planarity testing, convex hulls, network flows, and matching; the LEDA algorithmic library implemented many checkers for such problems [20]. As far as we could establish however, certifying algorithms have not been explicitly advocated or studied in the context of machine learning tasks.

Binary linear classification is one of the fundamental problems in supervised machine learning [22]. Linear and convex programming techniques have been used to approach binary classification since at least the 1960s [9, 17, 24], later giving rise to the idea of Support Vector Machines (SVMs) [6]. In such a context, the usefulness of convex duality theory is well-known [22, Section 5.2].

Kirchberger's theorem [13] was proved by mathematician Paul Kirchberger (a student of David Hilbert) in 1902, in the context of approximation theory. Several proofs of the theorem are known, all relating the theorem to results in combinatorial convexity and discrete geometry, such as Carathéodory's theorem and Helly's theorem [29, 30].

Carathéodory's theorem itself has seen several applications in linear and combinatorial optimization [4,21,26]. Closer to our setting, Haghighatkhah et al. [12] have used a constructive version of Carathéodory's theorem for bias removal in a classification context. Although the overarching goal of the algorithms is rather different, we reuse such a subroutine from [12, Section 2] in one of our algorithms. Our second algorithm is instead based on the Approximate Carathéodory theorem, the proof of which applies a probabilistic method [1, 28].

Organization In Section 2 we recall the main definitions and results relating to linear separability and combinatorial convexity, and we define the *Certified Linear Classification* problem. In Section 3 we present an algorithm for Certified Linear Classification and prove its correctness; this algorithm can be interpreted as a constructive and certifying extension of Kirchberger's theorem. In Section 4 we propose a dimension-free extension of Kirchberger's theorem, outline a corresponding algorithm and prove its correctness. We give some concluding remarks in Section 5.

2 Problem and Notation

Linear separability Let P and Q be two subsets in $\mathbb{R}^d$. Then P and Q are called (strictly) *linearly separable* if there exists a hyperplane H separating the two subsets, that is, if there exist a vector $v \in \mathbb{R}^d$ and constants $b, c \in \mathbb{R}$ ($c \neq 0$) such that

$$v^\top x + b \geq c \ , \text{ for all } x \in P \quad \text{and} \quad v^\top x + b \leq -c \ , \text{ for all } x \in Q.$$

Without loss of generality one can take $c = 1$ after rescaling. The above condition is then equivalent to the existence of $v \in \mathbb{R}^{d+1}$ such that

$$y v^\top \begin{pmatrix} x \\ 1 \end{pmatrix} \geq 1, \quad \text{for all } x \in P \cup Q, \tag{1}$$

where $y = 1$ whenever $x \in P$ and $y = -1$ whenever $x \in Q$.

Convexity We recall some notions from combinatorial convexity. Given a set P, the *convex hull* of $P \subseteq \mathbb{R}^d$ is the smallest convex set that contains P, and it is denoted as $CH(P)$. One can also characterize the convex hull $CH(P)$ as the set of every convex combination of all finite collections of points in P:

$$CH(P) = \left\{ \sum_{i=1}^{m} \lambda_i x^i \;\middle|\; m \in \mathbb{N}, \sum_{i=1}^{m} \lambda_i = 1 \text{ and } \forall\, 1 \leq i \leq m : x^i \in P,\ \lambda_i \geq 0 \right\}$$

The relevance of convex hulls for binary classification is due to the following observation (see for example [24, Section 2] for a proof).

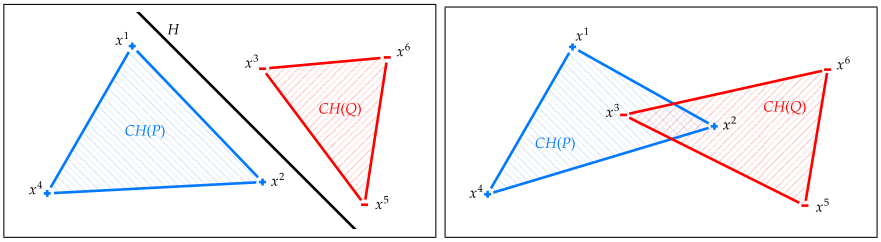

Fig. 1. Examples of linearly separable and inseparable sets. In the left subfigure, the convex hulls do not intersect; thus, it is possible to find a hyperplane H separating the two sets. On the other hand, in the right subfigure, the convex hulls have a non-empty intersection, making the sets linearly inseparable.

Remark 1. Two finite point sets P, Q are linearly separable if and only if $CH(P)$ and $CH(Q)$ do not intersect (see Figure 1 for an illustration).

The following results in combinatorial convexity will be useful. We use the notation $\mathrm{diam}(P)$ for the diameter of P, namely $\mathrm{diam}(P) \stackrel{\text{def}}{=} \sup_{x,x' \in P} \|x - x'\|_2$.

Theorem 1 (Carathéodory [7, 28]). *Every point in the convex hull of a set $P \subseteq \mathbb{R}^d$ can be expressed as a convex combination of at most $d+1$ points from P.*

Theorem 2 (Approximate Carathéodory [1, 28]). *Consider a set $P \subseteq \mathbb{R}^d$. Then, for every point $x \in CH(P)$ and every integer $k \geq 1$, one can find points $x^1, \ldots, x^k \in P$ such that*

$$\left\| x - \frac{1}{k} \sum_{j=1}^{k} x^j \right\|_2 \leq \frac{\mathrm{diam}(P)}{\sqrt{k}}. \tag{2}$$

Theorem 3 (Kirchberger [13, 30]). *Let P and Q be two finite sets of points in $\mathbb{R}^d$. Then, P and Q are linearly separable if and only if, for every set T of $d+2$ points chosen arbitrarily from $P \cup Q$, there exists a hyperplane separating $P \cap T$ and $Q \cap T$.*

Linear programming duality A *linear program* consists of the problem of minimizing a given linear function over the set of all vectors $v \in \mathbb{R}^d$ that satisfy a set of affine constraints, that is, a system of linear equalities and inequalities. Each linear program can be rewritten as

$$\begin{aligned} \texttt{minimize} \quad & b^\top v \\ \texttt{subject to} \quad & Av \geq c,\ v \geq 0 \end{aligned} \tag{3}$$

where A is a $n \times d$ real matrix, while b and c are vectors in $\mathbb{R}^d$ and $\mathbb{R}^n$, respectively. The problem defined in Eq. (3) will be called the *primal linear program*.

The *dual linear program* relies on maximizing a dual cost function:

$$\begin{aligned} \texttt{maximize} \quad & c^\top z \\ \texttt{subject to} \quad & A^\top z \leq b,\ z \geq 0. \end{aligned} \tag{4}$$

For further details on deriving the dual from its primal, see [18].

We refer to a vector $v \in \mathbb{R}^d$ satisfying all given constraints as a *feasible solution*. Moreover, each $v^* \in \mathbb{R}^d$ that minimizes the value $b^\top v$ over all feasible solutions is called an *optimal solution* or an *optimum*.

Theorem 4 (Strong Duality [18]). *For the primal and dual programs* (3) *and* (4)*, exactly one of the following holds:*

1. *Neither the primal nor the dual have a feasible solution.*
2. *The primal is unbounded (from below) and the dual has no feasible solution.*
3. *The primal has no feasible solution and the the dual is unbounded (from above).*
4. *Both the primal and the dual have a feasible solution. Then, both have an optimal solution. Furthermore, if v^* is an optimal solution of the primal and z^* is an optimal solution of the dual, then*

$$b^\top v^* = c^\top z^* \tag{5}$$

Certifying algorithm (informal definition) For the purposes of this work, an informal definition of certifying algorithm will be sufficient. A *(strongly) certifying algorithm* [3, 19] for a problem is, loosely speaking, an algorithm that produces, for each input x, either a witness w showing that x does not satisfy the problem's precondition, or an output y and a witness w showing that the pair (x, y) satisfies the problem's postcondition. The witness w is also called a *certificate*. The triple (x, y, w) can be forwarded to a *checker* C, another algorithm that accepts the triple if and only if w indeed proves that either x does not satisfy the precondition, or that (x, y) satisfies the postcondition. A fully formal definition of certifying algorithm can be found in [19, Section 5.1].

Problem definition Throughout this work, we assume to be working with n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$, for $i = 1, \ldots, n$. Furthermore, we assume, without loss of generality, that the given points satisfy $x^i_d = 1$; if not, we can simply add an additional dimension, with the extra coordinate of every point fixed to 1. This assumption is merely for notational convenience, as it allows to rewrite the condition of Eq. (1) as

$$y_i v^\top x^i \geq 1 \text{ for } i = 1, \ldots, n \tag{6}$$

where $v \in \mathbb{R}^d$.

Problem 1 (Certified Linear Classification).
Given: n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$.
Return: either a vector $v \in \mathbb{R}^d$ satisfying (6), or a subset of $d+2$ labeled points that no hyperplane can separate.

Example 1. Consider the sets $P = \{x^1, x^2, x^4\}$ and $Q = \{x^3, x^5, x^6\}$ depicted in Figure 1. In the left subfigure, the convex hulls $CH(P)$ and $CH(Q)$ do not intersect; hence, the sets are linearly separable, and a certifying algorithm may output a (properly scaled) vector normal to the hyperplane H. Conversely, the convex hulls of the sets in the right subfigure have a non-empty intersection, as the sets are linearly inseparable. A certifying algorithm thus may return any $d + 2 = 4$ points that cannot be linearly separated; for example, the subsets $P' = P$ and $Q' = \{x^3\}$.

A note on the computational model For the purpose of the running time analysis, we assume the unit-cost RAM model [2, 8].

3 A Certifying Algorithm

We now discuss a certifying algorithm for linear classification, the pseudo-code of which is given in Algorithm 1. The main intuition behind the algorithm is to exploit linear programming duality to construct, in the inseparable case, a point that acts as a negative witness using Remark 1.

We begin from a (standard) encoding of the separability problem as a linear optimization problem, which we call the primal problem.

$$\begin{aligned} \texttt{minimize} \quad & 0^\top v \\ \texttt{subject to} \quad & (YX)v \geq 1,\ v \in \mathbb{R}^d, \end{aligned} \tag{7}$$

Here, $X = (x^1 \cdots x^n)^\top$ is the $n \times d$ matrix of data points, and Y is the $n \times n$ diagonal matrix whose i-th entry is the label y_i of the i-th example x^i (recall that the last column of X consists entirely of ones since we assumed $x^i_d = 1$ for all data points). Any feasible solution v of problem (7) is a vector identifying a separating hyperplane, which can be interpreted as a positive certificate for the linear classification problem. The cost function is identically zero, but writing it as a linear function of v will be convenient for what follows.

To be able to deal with the negative (inseparable) case, we form the dual linear program to (7):

$$\begin{aligned} \texttt{maximize} \quad & 1^\top z \\ \texttt{subject to} \quad & (YX)^\top z = 0,\ z \in \mathbb{R}^n_{\geq 0} \end{aligned} \tag{8}$$

Applying strong duality to (7)–(8), we now get the following lemma.

Lemma 1. *Either both* (7) *and* (8) *have optimal solutions and the value of both programs is zero, or* (7) *is infeasible and* (8) *has a nonzero feasible solution.*

Proof. Observe that the set of feasible solutions for the dual problem in Eq. (8) is non-empty: indeed, the zero vector $0 \in \mathbb{R}^n$ always satisfies the constraints of the dual.

Algorithm 1: Certified Linear Classification

Input: n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$.
Output: a vector v satisfying Eq. (6), or a set of $d+2$ labeled points $P' \cup Q'$ that no hyperplane can separate.

$Y \leftarrow \mathbf{Diag}(y)$
Solve the primal-dual pair (7)–(8)
if *the primal has a solution* v^* **then**
  return *(`True`,* v^**)* `// Return a separating hyperplane`
else
  $z^* \leftarrow$ a nonzero dual solution `// Lemma 1`
  $I_{P'} \leftarrow \{i \in \{1, \ldots, n\} \mid y_i = +1\}$ `// Separate the indices`
  $I_{Q'} \leftarrow \{i \in \{1, \ldots, n\} \mid y_i = -1\}$
  $x^* \leftarrow \sum_{i \in I_{P'}} z_i^* x^i / \sum_{k \in I_{P'}} z_k^*$ `// Lemma 2`
  while $|I_{P'} \cup I_{Q'}| > d + 2$ **do** `// Reduce to` $d+2$ `points (Lemma 3)`
    if *there exists* i *such that* $z_i^* = 0$ **then**
      $I_{P'} \leftarrow I_{P'} \setminus \{i\}$, $I_{Q'} \leftarrow I_{Q'} \setminus \{i\}$
    else
      $\alpha \leftarrow$ a nonzero solution of Equations (16)-(18)
      $\rho_{P'} \leftarrow \min\{z_i^*/\alpha_i \mid i \in I_{P'},\ \alpha_i > 0\}$
      $\rho_{Q'} \leftarrow \min\{z_i^*/\alpha_i \mid i \in I_{Q'},\ \alpha_i > 0\}$
      $\rho \leftarrow \min\{\rho_{P'}, \rho_{Q'}\}$
      $z^* \leftarrow z^* - \rho\alpha$
      Find i such that $z_i^* = 0$
      $I_{P'} \leftarrow I_{P'} \setminus \{i\}$, $I_{Q'} \leftarrow I_{Q'} \setminus \{i\}$
    end
  end
  return *(`False`,* $I_{P'}$*,* $I_{Q'}$*,* x^**,* z^**)* `// Return a negative witness`
end

By strong duality (Theorem 4), since the dual has a feasible solution, we can deduce that either both programs have an optimal solution, or the primal has no feasible solution and the dual is unbounded: respectively, cases 3 and 4 in Theorem 4. If both programs have an optimal solution, the programs' value must be zero due to the primal objective and (5). On the other hand, if the dual is unbounded, a nonzero dual feasible solution must exist. □

By problem definition and Lemma 1, if both programs return an optimum, then the points $x^1, \ldots, x^n$ can be linearly separated. Furthermore, the vector v^* output as the primal solution is a normal vector associated with a separating hyperplane, which we can use as a positive certificate for linear separability.

Conversely, if the primal has no feasible solution, the points $x^1, \ldots, x^n$ are not linearly separable. Hence, we would like to extrapolate a negative certificate from the non-zero solution z^* obtained from the dual problem. Using Theorem 3, our goal will be to convert z^* into a subset of $d+2$ non-linearly-separable

points. This part of our algorithm is split into two subsequent steps, discussed in the following subsections.

3.1 Finding a point in the convex hulls' intersection

Suppose the dual is unbounded and we have computed a non-zero solution $z^* \in \mathbb{R}^n_{\geq 0}$ to the dual problem constraints in Eq. (8), meaning we found a non-zero and non-negative vector in the kernel of $(YX)^\top$. Since z^* is dual feasible,

$$(YX)^\top z^* = 0. \tag{9}$$

A non-zero vector z^* verifying Equation (9) can also be interpreted as a non-zero weight assignment to the points $x^1, \ldots, x^n$, each multiplied by its associated label y_i, such that the weighted sum is the zero vector:

$$(YX)^\top z^* = 0 \Leftrightarrow \sum_{i=1}^{n} z_i^* y_i x^i = 0. \tag{10}$$

Let P and Q be the sets containing the points with positive and negative labels, respectively. We can thus rewrite Equation (10) as

$$\sum_{i \in I_P} z_i^* x^i = \sum_{j \in I_Q} z_j^* x^j, \tag{11}$$

where we used I_P and I_Q to denote the sets of indices of elements in P and Q respectively. Since we are assuming $x_d^i = 1$ for all points, from the vector sums in Equation (11) we also deduce that the sum of the weights of elements in P equals the sum of weights of elements in Q, that is

$$\sum_{i \in I_P} z_i^* = \sum_{j \in I_Q} z_j^*. \tag{12}$$

Using Equation (12), we can divide both members of Equation (11) by their respective weights' sum, obtaining

$$\sum_{i \in I_P} \frac{z_i^*}{\sum_{k \in I_P} z_k^*} x^i = \sum_{j \in I_Q} \frac{z_j^*}{\sum_{k \in I_Q} z_k^*} x^j \tag{13}$$

Since $z^* \in \mathbb{R}^n_{\geq 0}$, the coefficients in both terms of Equation (13) are non-negative, and their sums are such that

$$\sum_{i \in I_P} \frac{z_i^*}{\sum_{k \in I_P} z_k^*} = \sum_{j \in I_Q} \frac{z_j^*}{\sum_{k \in I_Q} z_k^*} = 1. \tag{14}$$

Therefore, the left and right hand sums in Equation (13) are indeed two convex combinations in $CH(P)$ and $CH(Q)$, respectively. These convex combinations coincide and so they identify a point $x^* \in CH(P) \cap CH(Q)$, which, by Remark 1, already constitutes a *numerical* certificate of non-separability. We have shown the following.

Lemma 2. *If* (8) *has a non-zero solution* z^**, then the point*

$$x^* \stackrel{\text{def}}{=} \sum_{i \in I_P} \frac{z_i^*}{\sum_{k \in I_P} z_k^*} \, x^i \tag{15}$$

satisfies $x^* \in CH(P) \cap CH(Q)$. □

It is this intersection point x^* we are now going to use to compute the $d+2$ inseparable points composing the *combinatorial* part of the certificate.

3.2 Computing the Kirchberger points

To compute the $d+2$ points guaranteed by Kirchberger's theorem (Theorem 3), we use the following lemma and its proof, which expands on standard proofs of Caratheodory's theorem (Theorem 1); see Haghighatkhah et al. [12] for a similar algorithmic construction in a different context.

Lemma 3. *Let* P *and* Q *be two finite sets in* $\mathbb{R}^d$ *and* $x^* \in CH(P) \cap CH(Q)$*. Then one can find in time* $O(nd^3)$ *two subsets* $P' \subseteq P$ *and* $Q' \subseteq Q$ *such that* $x^* \in CH(P') \cap CH(Q')$ *and* $|P'| + |Q'| \leq d+2$*, where* $n = |P \cup Q|$*.*

Proof. Let $p \stackrel{\text{def}}{=} |P|$ and $q \stackrel{\text{def}}{=} |Q|$. If P and Q have a point in common, we can simply return that point, so assume that $P \cup Q = \{x^1, \ldots, x^{p+q}\}$; as before, we use $I_P, I_Q \subseteq \{1, \ldots, p+q\}$ to denote the sets of indices of points in P and Q, respectively. We show that if $p + q \geq d + 3$, then one point from either P or Q can be removed with the new convex hulls still intersecting. Let $x^* \in CH(P) \cap CH(Q)$. By definition of convex hull (Eq. (2)), there are coefficients $\lambda_1, \ldots, \lambda_{p+q} \geq 0$ such that $\sum_{i \in I_P} \lambda_i = 1$, $\sum_{j \in I_Q} \lambda_j = 1$ and

$$\sum_{i \in I_P} \lambda_i x^i = x^* = \sum_{j \in I_Q} \lambda_j x^j.$$

Now, there are two possibilities:

- If any of the λ_i coefficients is zero, then we can eliminate the corresponding point x^i while keeping x^* in the intersection of the convex hulls.
- If none of the λ_i coefficients is zero, we look for coefficients $\alpha_1, \ldots, \alpha_{p+q}$ such that

$$\sum_{i \in I_P} \alpha_i x^i = \sum_{j \in I_Q} \alpha_j x^j \tag{16}$$

$$\sum_{i \in I_P} \alpha_i = 0 \tag{17}$$

$$\sum_{j \in I_Q} \alpha_j = 0. \tag{18}$$

This linear system has $d+2$ equations and $p + q \geq d + 3$ variables. Hence, nonzero coefficients must exist that satisfy these equations.

Let $\rho_P = \min\{\lambda_i/\alpha_i \mid \alpha_i > 0, i \in I_P\}$, $\rho_Q = \min\{\lambda_j/\alpha_j \mid \alpha_j > 0, j \in I_Q\}$ and $\rho = \min\{\rho_P, \rho_Q\}$. Additionally, define the new coefficients as

$$\lambda'_i = \lambda_i - \rho\alpha_i, \quad \text{for } i = 1, \ldots, p+q. \tag{19}$$

By construction, we have $\lambda'_i \geq 0$ for $1 \leq i \leq p+q$, $\sum_{i\in I_P} \lambda'_i = \sum_{j\in I_Q} \lambda'_j = 1$, and $\sum_{i\in I_P} \lambda'_i x^i = \sum_{j\in I_Q} \lambda'_j x^j = x^*$. But at least one of the λ'_i coefficients is zero, and we can remove the corresponding point.

The above process can be repeated until $p+q = d+2$. Assuming we already have a point $x^* \in CH(P) \cap CH(Q)$ and its decomposition λ in terms of P and Q, finding the $d+2$ points relies on iteratively removing a point from $P \cup Q$ or solving a linear system of equations to eliminate a point and update the coefficients. The computational bottleneck resides in solving the linear system (16)–(18). This linear system can be defined using only $d+3$ arbitrarily chosen points from $P \cup Q$, and thus can be solved in time $O(d^3)$ using Gaussian elimination (of course, in a practical implementation, any fast linear solver can be used). Hence, computing the desired sets P' and Q' takes at most $O(nd^3)$ time, where $n = |P \cup Q|$. □

In summary, solving the dual problem (8) leads us to a point x^* in $CH(P) \cap CH(Q)$ and its decomposition in terms of elements of P and Q (Line 9 of Algorithm 1, justified by Lemma 2). From here on (Lines 10–22 of the algorithm), we follow the proof of Lemma 3 to eliminate elements from the convex combinations, while maintaining the invariant $x^* \in CH(P) \cap CH(Q)$. By the end of the computation, we are left with two reduced subsets P' and Q' that, by construction, cannot be linearly separated. The indices of points in P' and Q' are returned as a negative certificate, together with the point x^* and its decomposition (Line 23).

Theorem 5. *The Certified Linear Classification problem can be solved in time $O(n^{3.5}L + nd^3)$, where L is the input size.*

Proof. The primal and dual problems (problems (7)–(8)) can be efficiently solved using any primal-dual linear programming algorithm; interior-point algorithms applied to (8) run in $O(n^{3.5}L)$ time [14,23]. When P and Q are linearly separable, a separating hyperplane is identified without further computation needed. On the other hand, if P and Q are not linearly separable, we compute $x^* \in CH(P) \cap CH(Q)$ in linear time from the dual solution using (15) and then estimate the $d+2$ Kirchberger points from x^* following the procedure outlined in Lemma 3, which requires solving at most $n-d$ linear systems in $d+3$ variables and $d+2$ constraints each. □

3.3 Accompanying checker

The last piece to our algorithm is its accompanying checker. A checking algorithm should have a simple definition, and its running time should be lower than the

associated certifying algorithm [19]. Here we briefly outline a possible checker for Algorithm 1.

When Algorithm 1 returns `True`, it also returns a vector $v^* \in \mathbb{R}^d$ identifying a candidate separating hyperplane. To check correctness of this answer, the checker can simply test whether all points in P and Q satisfy Equation (6), by computing n inner products in $\mathbb{R}^d$. The checker thus runs in $O(nd)$ time (i.e. linear in the data size) when Algorithm 1 returns a `True` statement.

On the other hand, when Algorithm 1 returns `False`, it also returns the (at most) $d+2$ points in $P' \cup Q'$, the point x^* and the decomposition vector z^*. The checker can then simply verify that $z^* \geq 0$ (in $O(n)$ time) and that Equations from (11) to (15) hold; this proves that $x^* \in CH(P') \cap CH(Q') \subseteq CH(P) \cap CH(Q)$, which by Remark 1 certifies the inseparability of P and Q. Each of the equations can be checked in $O(d^2)$ time, by computing the appropriate linear combination of points. This is again linear in the data size as long as $d = O(n)$.

4 A Dimension-Free Extension

We now discuss an alternative certifying algorithm for linear separation based on the Approximate Carathéodory's theorem (Theorem 2). This approach may reduce the number of points in the negative certificate, as this number becomes independent of the ambient dimension d.

Let us assume once again to be working with n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$ and $x^i_d = 1$. Furthermore, we keep referring to P and Q as the sets of elements with positive and negative labels, respectively. Our goal is still to determine whether these n points can be separated by some hyperplane $H = \{x \in \mathbb{R}^d \mid v^\top x = 0\}$ where v satisfies Equation (6). Recall that the *margin* of a hyperplane H is the minimum distance between H and the data points:

$$\min_{x \in P \cup Q, x' \in H} \|x - x'\|_2 .$$

We note that the margin depends on the scale of the data and thus should be related to the diameter of the dataset, $D \stackrel{\text{def}}{=} \max\{\operatorname{diam}(P), \operatorname{diam}(Q)\}$. If a hyperplane has margin εD for some $\varepsilon > 0$, we will say that it has *normalized margin* ε with respect to the given dataset. To simplify exposition and without loss of generality, we assume that D is known. We remark that an approximation[1] of D within a factor of 2 can be computed in time linear in the data size.

If the points in P and Q can be separated (with *any* margin), our certifying algorithm will identify a separating hyperplane as a positive witness; otherwise, the algorithm will identify $O(1/\varepsilon^2)$ points (possibly with duplicates) that cannot be separated with normalized margin ε as a negative witness. In other words, the algorithm will solve the following problem.

[1] One can approximate e.g. $\operatorname{diam}(P)$ as $\tilde{D}_P \stackrel{\text{def}}{=} 2 \max_{x \in P} \|p - x\|_2$, after selecting an arbitrary $p \in P$, which satisfies $\operatorname{diam}(P) \leq \tilde{D}_P \leq 2 \operatorname{diam}(P)$ by the triangle inequality. Therefore, $D \leq \max\{\tilde{D}_P, \tilde{D}_Q\} \leq 2D$.

Problem 2 (Approximate Certified Linear Classification).
Parameter: $\varepsilon > 0$ (normalized margin).
Given: n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$.
Return: a vector $v \in \mathbb{R}^d$ satisfying Equation (6), or $O(1/\varepsilon^2)$ labeled points that no hyperplane can separate with normalized margin ε.

The main technical ingredient is the following.

Lemma 4. *Let P and Q be two finite sets in $\mathbb{R}^d$ and assume $x^* \in CH(P) \cap CH(Q)$. Then, for any $\varepsilon > 0$, there are two subsets $P' \subseteq P$ and $Q' \subseteq Q$ such that:*

1. *$|P'| = |Q'| = O(1/\varepsilon^2)$;*
2. *P' and Q' cannot be separated with normalized margin ε.*

Proof. By Theorem 2 applied to P (respectively, Q) and x^* with $k = \lceil 4/\varepsilon^2 \rceil$, there are k points $p'_1, \ldots p'_k$ in P (resp., $q'_1, \ldots, q'_k$ in Q) such that, if we define $x^+ = (1/k)\sum_i p'_i$ (resp., $x^- = (1/k)\sum_i q'_i$),

$$\begin{aligned} \left\|x^+ - x^*\right\|_2 &\leq \varepsilon D/2, \\ \left\|x^- - x^*\right\|_2 &\leq \varepsilon D/2. \end{aligned} \tag{20}$$

Therefore let us define $P' = \{p'_1, \ldots, p'_k\}$, $Q' = \{q'_1, \ldots, q'_k\}$. We note that by construction, $x^+ \in CH(P')$, $x^- \in CH(Q')$ (in fact, x^+ and x^- are the centroids of P' and Q' respectively). By (20) and the triangle inequality,

$$\left\|x^+ - x^-\right\|_2 \leq \varepsilon D. \tag{21}$$

But if P' and Q' were separable with normalized margin ε, the distance between any point in $CH(P')$ and any point in $CH(Q')$ would be at least $2\varepsilon D$, contradicting (21). Therefore the claim is proved. □

We give the full algorithm's pseudocode in Algorithm 2. The main steps are the following:

1. Find z^* by solving the primal-dual pair (7)–(8) (lines 1–6 of Algorithm 2);
2. Find $x^* \in CH(P) \cap CH(Q)$ and its convex decompositions in terms of points of P and Q using Equation (13): $x^* = \sum_i \lambda_i p_i$, $x^* = \sum_i \mu_i q_i$ (lines 7–11);
3. Sample $k \stackrel{\text{def}}{=} \lceil 16/\varepsilon^2 \rceil$ times the two probability distributions λ and μ, to obtain points $p'_1, \ldots, p'_k$ and $q'_1, \ldots, q'_k$ (lines 12–15).

The first two steps are exactly the same as in Algorithm 1, while the third step is different and is based on random sampling. Indeed, the points $p'_1, \ldots, p'_k$ and $q'_1, \ldots, q'_k$ referred to in the proof of Lemma 4 can be obtained as random samples (with replacement) from the sets P and Q according to the probability distributions that, interpreted as convex combinations, give rise to the point $x^* \in CH(P) \cap CH(Q)$. This idea is implicit in the proof of the Approximate Carathéodory theorem and is sometimes called the "empirical method" of B. Maurey [28, Chapter 0].

Algorithm 2: Approximate Certified Linear Classification

Input: $\varepsilon > 0$ and n points $x^1, \ldots, x^n$ in $\mathbb{R}^d$ with binary labels $y_i \in \{-1, 1\}$.
Output: a vector v satisfying Equation (6), or $O(1/\varepsilon^2)$ labeled points that no hyperplane can separate with normalized margin ε.

$Y \leftarrow \mathbf{Diag}(y)$
Solve the primal-dual pair (7)–(8)
if *the primal has a solution* v^* **then**
  return *(`True`,* v^**)* `// Return a separating hyperplane`
else
  $z^* \leftarrow$ a nonzero dual solution `// Lemma 1`
  $I_P \leftarrow \{i \in \{1, \ldots, n\} \mid y_i = +1\}$ `// Separate the indices`
  $I_Q \leftarrow \{i \in \{1, \ldots, n\} \mid y_i = -1\}$
  $x^* \leftarrow \sum_{i \in I_P} z_i^* x^i / \sum_{k \in I_P} z_k^*$ `// Lemma 2`
  $\lambda \leftarrow \{z_i^* / \sum_{j \in I_P} z_j^* \mid i \in I_P\}$
  $\mu \leftarrow \{z_i^* / \sum_{j \in I_Q} z_j^* \mid i \in I_Q\}$
  $k \leftarrow \lceil 16/\varepsilon^2 \rceil$
  $I_{P'} \leftarrow$ sequence of k indices sampled from distribution λ
  $I_{Q'} \leftarrow$ sequence of k indices sampled from distribution μ
  return *(`False`,* $I_{P'}$*,* $I_{Q'}$*)* `// Return a negative witness`
end

Theorem 6. *The Approximate Certified Linear Classification problem can be solved in randomized polynomial time.*

Proof. The only randomized step is the computation of the sets P' and Q'. It is enough to show that, with probability at least (say) $1/2$, the points $x^+ = (1/k)\sum_i p'_i$, $x^- = (1/k)\sum_i q'_i$ satisfy the inequalities (20); this probability can be increased as desired by increasing k. We take $k = \lceil 16/\varepsilon^2 \rceil$ (note that this is slightly larger than the value of k used in the proof of Lemma 4, which was only concerned with proving existence). Following the proof of the Approximate Carathéodory theorem as in [28, Chapter 0], we can bound

$$\begin{aligned} \mathbb{E}\left\|x^+ - x^*\right\|_2^2 &\leq \frac{D^2}{k}, \\ \mathbb{E}\left\|x^- - x^*\right\|_2^2 &\leq \frac{D^2}{k}. \end{aligned} \tag{22}$$

Therefore, by (22) and Markov's inequality, if $\gamma \stackrel{\text{def}}{=} \varepsilon D/2$,

$$\Pr[\left\|x^+ - x^*\right\|_2 \geq \gamma] = \Pr[\left\|x^+ - x^*\right\|_2^2 \geq \gamma^2] \leq \frac{D^2}{k\gamma^2} = \frac{4}{k\varepsilon^2} \leq \frac{1}{4},$$

and similarly $\Pr[\|x^- - x^*\|_2 \geq \gamma] \leq 1/4$. Thus, by a union bound, (20) is satisfied with probability at least $1/2$. The rest of the argument proceeds as in the proof of Lemma 4 and allows us to conclude that P' and Q' cannot be separated with normalized margin ε. □

Finally, the associated checker to Algorithm 2 retraces the strategy of the checker presented in Subsection 3.3 in the positive case. In the negative case, on the other hand, one only need compute the centroids of the two sets P' and Q' and check that their distance is at most εD. This requires $O(d/\varepsilon^2)$ time and $O(d)$ time, respectively. Combining the positive and negative cases, the running time of the checker is $O(nd + d/\varepsilon^2)$.

5 Conclusions

In this work we proposed to apply the idea of certifying algorithms to machine learning problems, and we focused on one of the fundamental tasks in this context, binary linear classification. We have shown that both the Certified Linear Classification problem and the Approximate Certified Linear Classification problem can be solved efficiently and that they admit simple checkers. The running time of the checker for Algorithm 1 is at most linear in the data size as long as $d = O(n)$, and the running time of the checker for Algorithm 2 is at most linear in the data size as long as $1/\varepsilon^2 = O(n)$. It would be interesting to study the Certified Linear Classification problem in a high-dimensional regime where $d \gg n$. It would also be interesting to design certifying algorithms for multi-class classification.

More generally, there are clearly other relevant machine learning problems, such as clustering, that one could also investigate from the point of view of certifying algorithms; we hope that this work can stimulate further research in this area.

Acknowledgments. This research was partly supported by project ECS 0000024 of the European Commission, *Rome Technopole*, PNRR grant M4-C2-Inv. 1.5. In particular, manuscript writing and editing were funded by Rome Technopole. The authors would also like to thank the anonymous reviewers for their feedback.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Adiprasito, K., Bárány, I., Mustafa, N.H., Terpai, T.: Theorems of Carathéodory, Helly, and Tverberg without dimension. Discrete & Computational Geometry **64**(2), 233–258 (2020). https://doi.org/10.1007/s00454-020-00172-5
2. Aho, A.V., Hopcroft, J.E., Ullman, J.D.: The Design and Analysis of Computer Algorithms. Addison-Wesley (1974)
3. Alkassar, E., Böhme, S., Mehlhorn, K., Rizkallah, C., Schweitzer, P.: An introduction to certifying algorithms. it - Information Technology **53**(6), 287–293 (Dec 2011). https://doi.org/10.1524/itit.2011.0655
4. Barman, S.: Approximating Nash equilibria and dense subgraphs via an approximate version of Carathéodory's theorem. SIAM J. Comput. **47**(3), 960–981 (2018). https://doi.org/10.1137/15M1050574

5. Blum, M., Kannan, S.: Designing programs that check their work. J. ACM **42**(1), 269–291 (1995). https://doi.org/10.1145/200836.200880
6. Boser, B.E., Guyon, I., Vapnik, V.: A training algorithm for optimal margin classifiers. In: Haussler, D. (ed.) Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh, PA, USA, July 27-29, 1992. pp. 144–152. ACM (1992). https://doi.org/10.1145/130385.130401
7. Carathéodory, C.: Über den Variabilitätsbereich der Koeffizienten von Potenzreihen, die gegebene Werte nicht annehmen. Mathematische Annalen **64**(1), 95–115 (1907). https://doi.org/10.1007/BF01449883
8. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 3rd Edition. MIT Press (2009), http://mitpress.mit.edu/books/introduction-algorithms
9. Cover, T.M.: Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Trans. Electron. Comput. **14**(3), 326–334 (1965). https://doi.org/10.1109/PGEC.1965.264137
10. Dhiflaoui, M., Funke, S., Kwappik, C., Mehlhorn, K., Seel, M., Schömer, E., Schulte, R., Weber, D.: Certifying and repairing solutions to large LPs – How good are LP-solvers? In: Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA. pp. 255–256. ACM/SIAM (2003), http://dl.acm.org/citation.cfm?id=644108.644152
11. Georgiadis, L., Tarjan, R.E.: Dominator tree certification and divergent spanning trees. ACM Trans. Algorithms **12**(1), 11:1–11:42 (2016). https://doi.org/10.1145/2764913, https://doi.org/10.1145/2764913
12. Haghighatkhah, P., Meulemans, W., Speckmann, B., Urhausen, J., Verbeek, K.: Obstructing Classification via Projection. In: Bonchi, F., Puglisi, S.J. (eds.) 46th International Symposium on Mathematical Foundations of Computer Science (MFCS 2021). Leibniz International Proceedings in Informatics (LIPIcs), vol. 202, pp. 56:1–56:19. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). https://doi.org/10.4230/LIPIcs.MFCS.2021.56
13. Kirchberger, P.: Über Tchebychefsche Annäherungsmethoden. Mathematische Annalen **57**, 509–540 (1903). https://doi.org/10.1007/BF01445182
14. Kojima, M., Mizuno, S., Yoshise, A.: An $O(\sqrt{n}L)$ iteration potential reduction algorithm for linear complementarity problems. Math. Program. **50**, 331–342 (1991). https://doi.org/10.1007/BF01594942
15. Kratsch, D., McConnell, R.M., Mehlhorn, K., Spinrad, J.P.: Certifying algorithms for recognizing interval graphs and permutation graphs. SIAM J. Comput. **36**(2), 326–353 (2006). https://doi.org/10.1137/S0097539703437855
16. Lipton, Z.C.: The mythos of model interpretability. Commun. ACM **61**(10), 36–43 (2018). https://doi.org/10.1145/3233231
17. Mangasarian, O.L.: Linear and nonlinear separation of patterns by linear programming. Operations Research **13**(3), 444–452 (1965), https://www.jstor.org/stable/167808
18. Matoušek, J., Gärtner, B.: Understanding and Using Linear Programming. Springer (2007)
19. McConnell, R.M., Mehlhorn, K., Näher, S., Schweitzer, P.: Certifying algorithms. Computer Science Review **5**(2), 119–161 (2011). https://doi.org/10.1016/j.cosrev.2010.09.009
20. Mehlhorn, K., Näher, S.: LEDA: A Platform for Combinatorial and Geometric Computing. Cambridge University Press (1999), http://www.mpi-sb.mpg.de/%7Emehlhorn/LEDAbook.html

21. Mirrokni, V.S., Leme, R.P., Vladu, A., Wong, S.C.: Tight bounds for approximate Carathéodory and beyond. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017. Proceedings of Machine Learning Research, vol. 70, pp. 2440–2448. PMLR (2017), http://proceedings.mlr.press/v70/mirrokni17a.html
22. Mohri, M., Rostamizadeh, A., Talwalkar, A.: Foundations of Machine Learning. Adaptive computation and machine learning, MIT Press (2012)
23. Monteiro, R.D.C., Adler, I.: Interior path following primal-dual algorithms. Part I: Linear programming. Math. Program. **44**(1-3), 27–41 (1989). https://doi.org/10.1007/BF01587075
24. Rosen, J.B.: Pattern separation by convex programming. Journal of Mathematical Analysis and Applications **10**(1), 123–134 (1965). https://doi.org/10.1016/0022-247X(65)90150-2
25. Schmidt, J.M.: Contractions, removals, and certifying 3-connectivity in linear time. SIAM J. Comput. **42**(2), 494–535 (2013). https://doi.org/10.1137/110848311, https://doi.org/10.1137/110848311
26. Schrijver, A.: Combinatorial Optimization. Springer (2004)
27. Seshia, S.A., Sadigh, D., Sastry, S.S.: Toward verified artificial intelligence. Commun. ACM **65**(7), 46–55 (2022). https://doi.org/10.1145/3503914
28. Vershynin, R.: High-Dimensional Probability: An Introduction with Applications in Data Science. Cambridge Series in Statistical and Probabilistic Mathematics, Cambridge University Press (2018). https://doi.org/10.1017/9781108231596
29. Watson, D.: A refinement of theorems of Kirchberger and Carathéodory. Journal of the Australian Mathematical Society **15**(2), 190–192 (1973). https://doi.org/10.1017/S1446788700012957
30. Webster, R.J.: Another simple proof of Kirchberger's theorem. Journal of Mathematical Analysis and Applications **92**(1), 299–300 (1983). https://doi.org/10.1016/0022-247X(83)90286-X