Knowledge and Information Systems
https://doi.org/10.1007/s10115-024-02108-4

REGULAR PAPER

Robustness verification of k-nearest neighbors by abstract interpretation

Nicolò Fassina[1] · **Francesco Ranzato**[1] · **Marco Zanella**[1]

Received: 4 January 2024 / Revised: 6 March 2024 / Accepted: 21 March 2024

Abstract
We study the certification of stability properties, such as robustness and individual fairness, of the k-nearest neighbor algorithm (kNN). Our approach leverages abstract interpretation, a well-established program analysis technique that has been proven successful in verifying several machine learning algorithms, notably, neural networks, decision trees, and support vector machines. In this work, we put forward an abstract interpretation-based framework for designing a sound approximate version of the kNN algorithm, which is instantiated to the interval and zonotope abstractions for approximating the range of numerical features. We show how this abstraction-based method can be used for stability, robustness, and individual fairness certification of kNN. Our certification technique has been implemented and experimentally evaluated on several benchmark datasets. These experimental results show that our tool can formally prove the stability of kNN classifiers in a precise and efficient way, thus expanding the range of machine learning models amenable to robustness certification.

Keywords k-nearest neighbors · Robustness · Individual fairness · Data poisoning · Formal certification

1 Introduction

k-nearest neighbors (kNN) [2] are one of the simplest supervised machine learning (ML) algorithms. Nevertheless, kNN is a popular and accurate predictive model with diverse application fields [21]. The basic idea of kNN is to predict the outcome for an input sample $\mathbf{x} \in \mathbb{R}^n$ by inferring the k nearest neighbors of $\mathbf{x}$ ranging in a given dataset. The number $k \in \mathbb{N}$ of neighbors as well as the distance function between vectors is parameters of this model. Once the set of k nearest neighbors of an input sample is computed, the output is inferred as the most common label of these k neighbors in case of classification, or as average of the values of the k neighbors in case of regression. The diagram in Fig. 1 depicts an example of classification for a kNN model with $k = 3$ and a dataset in $\mathbb{R}^2$ with three classes *red*, *green*, and *blue*. For an input vector $\mathbf{x}$ represented by a black bullet, 3NN therefore computes the

✉ Francesco Ranzato
francesco.ranzato@unipd.it

[1] Dipartimento di Matematica, University of Padova, Via Trieste 63, 35121 Padua, Italy

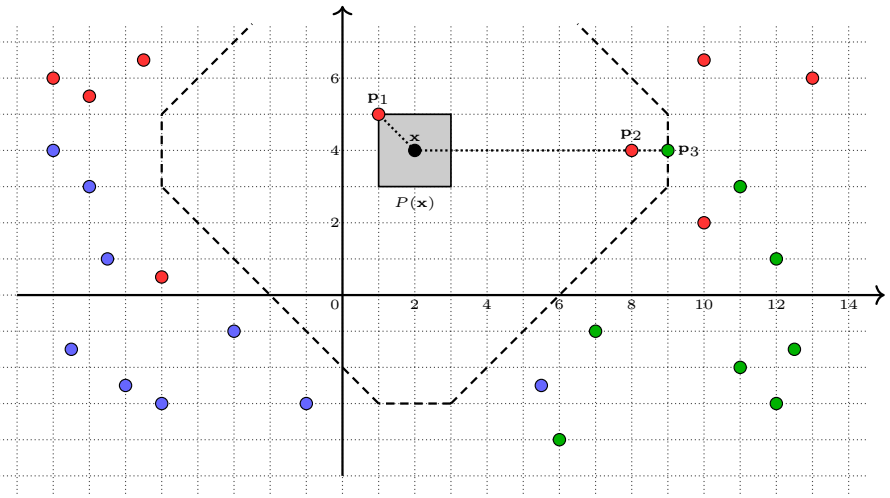

Fig. 1 *k*NN on a dataset with three classes red, green, blue (color figure online)

3 nearest samples in the dataset w.r.t. Manhattan distance, as depicted by the dashed lines, and then the most common label among them is inferred. In *k*NN, the dataset is stored and entirely used at classification time; namely, *k*NN is a lazy (or "just-in-time") learning algorithm [4]. While this makes *k*NN simple to implement, it can exhibit a significant prediction time due to the computational effort required to calculate distances for the whole dataset and, correspondingly, for sorting samples, especially for high values of k. (k is usually a low odd value, often below 9.)

Adversarial machine learning [18, 22, 28] studies vulnerabilities of ML in adversarial scenarios. Adversarial examples have been found in diverse application fields of ML, and the current defense techniques include adversarial model training, input validation, testing and automatic formal certification of learning algorithms. A ML classifier C is defined to be *stable* on an input $\mathbf{x}$ for a (typically very small) perturbation $P(\mathbf{x})$ of $\mathbf{x}$ which represents an adversarial attack, when C assigns the same class to all the samples in $P(\mathbf{x})$. Moreover, when such class is also the correct class of $\mathbf{x}$ with respect to ground truth, the classifier C is *robust* on $\mathbf{x}$ as it cannot be deceived by unnoticeable malicious alterations of $\mathbf{x}$. Figure 1 depicts in gray an adversarial region $P(\mathbf{x})$ defined around the black input sample $\mathbf{x}$, which represents an (infinite) set of attacks. Here, the 3 nearest neighbors of each attack in $P(\mathbf{x})$ are labeled as *red* (being $\mathbf{p}_1$ and $\mathbf{p}_2$) and *green* (being $\mathbf{p}_3$), making 3NN stable on $\mathbf{x}$ as 3NN classifies $\mathbf{x}$ are *red*. If *red* is the ground truth label for $\mathbf{x}$, then 3NN is robust on $\mathbf{x}$ as well.

1.1 Contributions

Our main contribution is a novel formal and automatic verification method for inferring when a *k*NN classifier is *provably stable* for an input sample with respect to a given perturbation. We leverage the well-established framework of abstract interpretation [7, 8, 17] for computing correct over-approximations of dynamic system behaviors, which has already been successfully applied to the formal verification of diverse machine learning models (see the surveys [1, 26, 44]). Our approach is based on designing a sound abstract version $C^A_{\delta,k}$ of a *k*NN classifier based on a distance function δ, e.g., Euclidean or Manhattan distance. This approx-

imate classifier $C^A_{\delta,k}$ is defined over a symbolic numerical abstraction A of the input space $\wp(\mathbb{R}^n)$, and leverages a sound approximation δ^A in A of the distance function δ. In turn, the definition of δ^A relies on sound approximations over A of its basic numerical operations such as addition, product, and modulus. Given an abstract value $a \in A$ which provides a symbolic over-approximation of an adversarial perturbation $P(\mathbf{x})$ of an input sample $\mathbf{x}$, $C^A_{\delta,k}(a)$ returns an over-approximation of the set of classes computed by kNN for all the samples in $P(\mathbf{x})$. Hence, if $C^A_{\delta,k}(a) = k\text{NN}(\mathbf{x})$ holds, then we can infer that kNN is provably stable on $\mathbf{x}$ for its perturbation $P(\mathbf{x})$. We instantiate our certification method to the well-known numerical abstract domains of intervals [8] and zonotopes [19], that approximate the range of numerical features by, resp., real intervals (e.g., $\mathbf{x}_i \in [l, u]$) and affine forms (e.g., $\mathbf{x}_i = a_0 + \sum_{j=1}^{k} a_j \epsilon_j$ with $a_j \in \mathbb{R}$ and noise symbols $\epsilon_j \in [-1, 1]$). This certification framework for kNN has been implemented in Python. The corresponding tool, called NAVe (kNN Abstract Verifier; the Italian word "nave" means "ship"), has been designed to be scalable both in the size of the training dataset and in the value of k, for which no upper bound is assumed. We performed an experimental evaluation of NAVe on seven datasets commonly used in robustness certification and on two additional datasets for individual fairness verification. These experimental results show that NAVe is an effective tool for formally certifying the adversarial robustness of inputs to kNN, and that, in general, kNN turns out to be a quite robust prediction algorithm: In fact, for adversarial perturbations $\leq \pm 2\%$, NAVe is able to infer for several datasets more than 90% of robustness for $k \in \{1, 3, 5, 7\}$.

1.2 Illustrative example

Let us consider the example in $\mathbb{R}^2$ depicted in Fig. 1, where $\mathbf{x} = (2, 4)$ is the input sample and $P(\mathbf{x}) \triangleq \{\mathbf{x}' \in \mathbb{R}^2 \mid \max(|\mathbf{x}'_1 - \mathbf{x}_1|, |\mathbf{x}'_2 - \mathbf{x}_2|) \leq 1\}$ is a perturbation defined as the ℓ_∞ ball of radius 1 centered in $\mathbf{x}$, which can be exactly represented through intervals as $(\mathbf{x}_1 \in [1, 3], \mathbf{x}_2 \in [3, 5])$. By leveraging the interval abstract domain $\mathcal{I}$, we compute the abstract Manhattan distance $\mu^{\mathcal{I}}$ between $P(\mathbf{x})$ and the 3 points $\mathbf{p}_1 = (1, 5)$, $\mathbf{p}_2 = (8, 4)$, $\mathbf{p}_3 = (9, 4)$ of the training dataset:

$$\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_1) = |[1, 3] -^{\mathcal{I}} 1|^{\mathcal{I}} +^{\mathcal{I}} |[3, 5] -^{\mathcal{I}} 5|^{\mathcal{I}} = [0, 2] +^{\mathcal{I}} [0, 2] = [0, 4]\,,$$
$$\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_2) = |[1, 3] -^{\mathcal{I}} 8|^{\mathcal{I}} +^{\mathcal{I}} |[3, 5] -^{\mathcal{I}} 4|^{\mathcal{I}} = [5, 7] +^{\mathcal{I}} [0, 1] = [5, 8]\,,$$
$$\mu^{\mathcal{I}}(P(\mathbf{x}), \mathbf{p}_3) = |[1, 3] -^{\mathcal{I}} 9|^{\mathcal{I}} +^{\mathcal{I}} |[3, 5] -^{\mathcal{I}} 4|^{\mathcal{I}} = [6, 8] +^{\mathcal{I}} [0, 1] = [6, 9]\,.$$

These abstract distances are symbolically computed in the interval abstraction $\mathcal{I}$ and provide correct lower and upper bounds for the infinite set of Manhattan distances

$$\{\mu(\mathbf{y}, \mathbf{p}_i) \in \mathbb{R}_{\geq 0} \mid \mathbf{y} \in P(\mathbf{x})\}.$$

By leveraging these abstract distances, for any number of neighbors $k \in \mathbb{N}^*$, the abstract classifier $C^{\mathcal{I}}_{\mu,k}(P(\mathbf{x}))$ returns an over-approximation of the set of classes $\cup_{\mathbf{y} \in P(\mathbf{x})} k\text{NN}(\mathbf{y})$. Let us observe that $\mathbf{p}_1$ is the nearest point to $P(\mathbf{x})$, as its interval $[0, 4]$ is strictly dominated by all the others. ($[l_1, u_1]$ is strictly dominated by $[l_2, u_2]$ when $u_1 < l_2$.) As a consequence, $C^{\mathcal{I}}_{\mu,1}(P(\mathbf{x})) = \{red\}$, so that we proved that 1NN is stable on $\mathbf{x}$. On the other hand, it turns out that $\mathbf{p}_2$ is closer than $\mathbf{p}_3$ to every point in $P(\mathbf{x})$, although this cannot be inferred from the corresponding abstract distances since the interval $[5, 8]$ for $\mathbf{p}_2$ is not strictly dominated by $[6, 9]$ for $\mathbf{p}_3$: This is an example of loss of precision, also called *incompleteness* of the stability certification. Consequently, if we use $k = 2$ in this scenario, then we cannot exclude $\mathbf{p}_3$ by the approximate set of neighbors, which could be either $\{\mathbf{p}_1, \mathbf{p}_2\}$, thus resulting in a

red output, or $\{\mathbf{p}_1, \mathbf{p}_3\}$, thus causing an ambiguity between a *red* or *green* output. This entails that $C^{\mathcal{T}}_{\mu,2}(P(\mathbf{x})) = \{red, green\}$, so that stability of 2NN on $\mathbf{x}$ cannot be proved. In this case, *green* is therefore a false positive, arising from the interval approximation. Finally, for $k = 3$, the stability verification turns out to be complete, because the three samples $\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3$ are the unique points which will be taken into account, as hinted by the black dashed line in Fig. 1, so that $C^{\mathcal{T}}_{\mu,3}(P(\mathbf{x})) = \{red\}$ holds, thus allowing us to infer that 3NN is stable on $\mathbf{x}$.

1.3 Related work

Formal verification methods in adversarial machine learning have been thoroughly investigated for (deep) neural networks, while different ML models have been much less studied. In particular, adversarial attacks on k-nearest neighbor algorithms have been studied only recently [3, 11, 20, 23–25, 41, 42, 45, 46, 48]. Among them, let us mention [42], where the authors put forward an algorithm, called GeoAdEx, based on higher-order Voronoi diagrams, that aims at finding the smallest perturbation that moves an input sample to an adversarial cell, which is an order-k Voronoi cell that has a different majority label. However, finding this smallest perturbation, or a certified lower bound for it, may often need a long time, essentially due to a combinatorial complexity, so that in most cases GeoAdEx outputs exact results, i.e., without approximations, only for $k = 1$. Moreover, Fan et al. [11]'s approach is orthogonal to ours: (i) Their notion of robustness is different, since an input $\mathbf{x}$ is considered to be robust w.r.t. a set of datasets $\mathcal{I}$, when there exists a label l such that for all $D \in \mathcal{I}$, $k\mathrm{NN}_D(\mathbf{x}) = l$; (ii) [11] studies the theoretical complexity of certifying this different concept of robustness w.r.t. a notion of subset repair of datasets. Let us finally mention that [23, 25] prove robustness of kNN to adversarial poisoning of the dataset by leveraging an over-approximated kNN classifier, while [24] puts forward an abstraction-based method for certifying the fairness of kNN under the assumption that the training data may have bias caused by systematic mislabeling of samples. While these works [23–25] leverage some specific sound over-approximations of the procedures involved in kNN classification, they are not firmly designed and specified within the compositional abstract interpretation framework [7, 8]; namely, they are not parametric on some underlying numerical abstract domains (such as the interval and zonotope abstractions employed in this work) and on the corresponding abstract operations (such as abstract additions, exponentials and modulus) to be used for defining abstract distances. Abstract interpretation techniques have been applied for designing precise and scalable robustness verification algorithms and adversarial training techniques for a range of ML models [5, 15, 27, 32–37, 39, 40]. To the best of our knowledge, no prior work applied abstract interpretation for the robustness certification of k-nearest neighbors.

This article is a full and revised version of the ICDM2023 conference paper [13], extended to include all the technical proofs and the following novel contributions: Sect. 2.1.2 introduces a new sound abstraction of the modulus operation on zonotopes; Sect. 3.3 shows how to extend the verification method to regression tasks; Sect. 3.4 discusses how different abstractions and perturbations can be used in our approach; and Sect. 4 studies the relationship between our notion of stability with data poisoning.

2 Background

2.1 Numerical abstract domains

A numerical abstract domain (or numerical abstraction) [31] A symbolically represents sets of real vectors through a so-called concretization map $\gamma^A : A \to \wp(\mathbb{R}^n)$ providing the meaning of its abstract (i.e., symbolic) values. A subset of vectors $S \in \wp(\mathbb{R}^n)$ is over-approximated by some abstract value $a \in A$ when $S \subseteq \gamma^A(a)$, while S is exactly represented by a when $S = \gamma^A(a)$ holds. An abstract domain A may also admit an abstraction function $\alpha^A : \wp(\mathbb{R}^n) \to A$ such that $\alpha^A(S)$ is the best abstraction in A of the set S, where the notion of best means least (or minimal) w.r.t. the following preorder relation on A: $a \sqsubseteq^A a' \Leftrightarrow \gamma^A(a) \subseteq \gamma^A(a')$. If $\langle A, \sqsubseteq^A \rangle$ is a partially ordered set, then the concretization and abstraction maps form a Galois connection: For all $S \in \wp(\mathbb{R}^n)$ and $a \in A$, $\alpha^A(S) \sqsubseteq^A a \Leftrightarrow S \subseteq \gamma^A(a)$ holds.

Given a k-ary operation on vectors $f : (\mathbb{R}^n)^k \to \mathbb{R}^n$, for some $k \geq 1$, an abstract function $f^A : A^k \to A$ is a sound (or correct) (over-)approximation of f when for all $(a_1, \ldots, a_k) \in A^k$, the containment

$$\{f(\mathbf{x}_1, \ldots, \mathbf{x}_k) \mid \forall i \cdot \mathbf{x}_i \in \gamma^A(a_i)\} \subseteq \gamma^A(f^A(a_1, \ldots, a_k))$$

holds, while f^A is defined to be exact (or complete) when equality holds. In words, soundness holds when $f^A(a_1, \ldots, a_k)$ never misses a concrete computation of f on some input $(\mathbf{x}_1, \ldots, \mathbf{x}_k)$ which is abstractly represented by $(a_1, \ldots, a_k)$, while exactness means that each abstract computation $f^A(a_1, \ldots, a_k)$ is an exact abstract representation of the set of concrete computations of f on all the inputs that are abstractly represented by $(a_1, \ldots, a_k)$. If A is endowed with an abstraction map α^A, then the function

$$f^A_{\text{best}} \triangleq \lambda(a_1, \ldots, a_k) \cdot \alpha^A(f(\gamma^A(a_1), \ldots, \gamma^A(a_k)))$$

is called the best correct approximation of f, because for any other correct approximation f^A, $f^A_{\text{best}}(a_1, \ldots, a_k) \sqsubseteq^A f^A(a_1, \ldots, a_k)$ always holds. Thus, f^A_{best} represents the best possible approximation of f that can be defined on the abstract domain A.

Intervals The abstract domain of real intervals $\mathcal{I}$ is one of the simplest and most used abstractions in ML verification. The interval domain abstracts the values of a real variable by a (possibly unbounded) real interval $[l, u]$, where $l, u \in \mathbb{R} \cup \{-\infty, +\infty\}$ and $l \leq u$ (with $-\infty \leq x \leq +\infty$ for all $x \in \mathbb{R}$). Moreover, $\mathcal{I}$ includes a symbolic representation $\bot^{\mathcal{I}}$ of the empty set. The concretization $\gamma^{\mathcal{I}} : \mathcal{I} \to \wp(\mathbb{R})$ is defined as follows: $\gamma^{\mathcal{I}}(\bot^{\mathcal{I}}) \triangleq \varnothing$; $\gamma^{\mathcal{I}}([l, u]) \triangleq \{x \in \mathbb{R} \mid l \leq x \leq u\}$. The product interval abstraction $\mathcal{I}^n$, with $n \geq 1$, is also called the box (or hyperrectangle) domain, and its concretization map $\gamma^{\mathcal{I}^n} : \mathcal{I}^n \to \wp(\mathbb{R}^n)$ is defined by a straightforward componentwise product of $\gamma^{\mathcal{I}}$. Intervals have an abstraction map $\alpha^{\mathcal{I}} : \wp(\mathbb{R}) \to \mathcal{I}$ which is defined as follows:

$$\alpha^{\mathcal{I}}(X) \triangleq \begin{cases} \bot^{\mathcal{I}} & \text{if } X = \emptyset \\ [\inf X, \sup X] & \text{otherwise} \end{cases}$$

Zonotopes The interval domain can be imprecise as it is nonrelational, i.e., $\mathcal{I}$ does not represent information on how values of different variables are related. For example, the most precise interval approximation of the set $T = \{(x, y) \in \mathbb{R}^2 \mid 0 \leq x, y \leq 1, x = y\}$ is $\langle x \in [0, 1], y \in [0, 1] \rangle$, thus losing the information that $x - y = 0$. The zonotope abstract domain $\mathcal{Z}$ [16, 19] is based on affine arithmetic [9] and can be viewed as an extension of

intervals that keeps track of affine relations between values of different variables. The domain $\mathcal{Z}$ consists of abstract values $\hat{a} = a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathcal{Z}$, where $a_j \in \mathbb{R}$ are coefficients and ϵ_j are noise symbols whose values range in the real interval $[-1, 1]$, and when these ϵ_j are shared between different variables/features, they encode a relation between them. The concretization of a zonotope $\hat{a}$ is given by

$$\gamma^{\mathcal{Z}}(\hat{a}) \triangleq \left\{ a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathbb{R} \mid \forall j \cdot \epsilon_j \in [-1, 1] \right\},$$

i.e., the zonotope $\hat{a}$ represents the real interval $\left[a_0 - \sum_{j=1}^{m} |a_j|, a_0 + \sum_{j=1}^{m} |a_j|\right]$. The product zonotope abstraction $\mathcal{Z}^n$, with $n \geq 1$, may share noise symbols between different components, thus enabling to represent relational information between features. For example, the above set $T \subseteq \mathbb{R}^2$ can be exactly represented by the zonotope $(\hat{x} = 0.5 + 0.5\epsilon_1, \hat{y} = 0.5 + 0.5\epsilon_1)$, so that we can infer that $\hat{x} - \hat{y} = 0$ holds. A fundamental property of zonotopes is that linear functions, such as vector addition and constant multiplication, admit corresponding exact abstract operations on $\mathcal{Z}$, while nonaffine functions, such as multiplications and modulus, must necessarily be approximated.

The basic abstract operations on intervals and zonotopes for computing abstract distances are recalled below.

2.1.1 Abstract operations on intervals

The most precise abstract operations, that is, the best correct approximations, on $\mathcal{I}$ are well known [31] and recalled below.

addition: $[l_1, u_1] +^{\mathcal{I}} [l_2, u_2] \triangleq [l_1 + l_2, u_1 + u_2]$

constant multiplication: $c[l, u] \triangleq \begin{cases} [cl, cu] & \text{if } c \geq 0 \\ [cu, cl] & \text{otherwise} \end{cases}$

multiplication: $[l_1, u_1] \cdot^{\mathcal{I}} [l_2, u_2] \triangleq [\min(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2), \max(l_1 l_2, l_1 u_2, u_1 l_2, u_1 u_2)]$

modulus: $|[l, u]|^{\mathcal{I}} \triangleq \begin{cases} [\min(|l|, |u|), \max(|l|, |u|)] & \text{if } lu \geq 0 \\ [0, \max(|l|, |u|)] & \text{otherwise} \end{cases}$

exponential: $[l, u]^{p^{\mathcal{I}}} \triangleq \begin{cases} [l^p, u^p] & \text{if } p \text{ odd or } l \geq 0 \\ [u^p, l^p] & \text{if } p \text{ even and } u < 0 \\ [0, \max(l^p, u^p)] & \text{otherwise} \end{cases}$

dominance test: $[l_1, u_1] <^{\mathcal{I}} [l_2, u_2] \triangleq u_1 < l_2$

In particular, soundness of the dominance test means that if $[l_1, u_1] <^{\mathcal{I}} [l_2, u_2]$, then for all $x \in \gamma^{\mathcal{I}}([l_1, u_1])$ and $y \in \gamma^{\mathcal{I}}([l_2, u_2])$, $x < y$ holds.

2.1.2 Abstract operations on zonotopes

Zonotopes are exact for linear operations, namely addition and constant multiplication, while for nonlinear operations, in particular multiplication and modulus, the result, in general, cannot be exactly represented by a zonotope, so that the multiplication of zonotopes approximates the precise result by adding a fresh noise symbol ϵ_f whose coefficient is typically computed by a Taylor approximation of the nonlinear part of the multiplication (see [19, Section 2.1.5]).

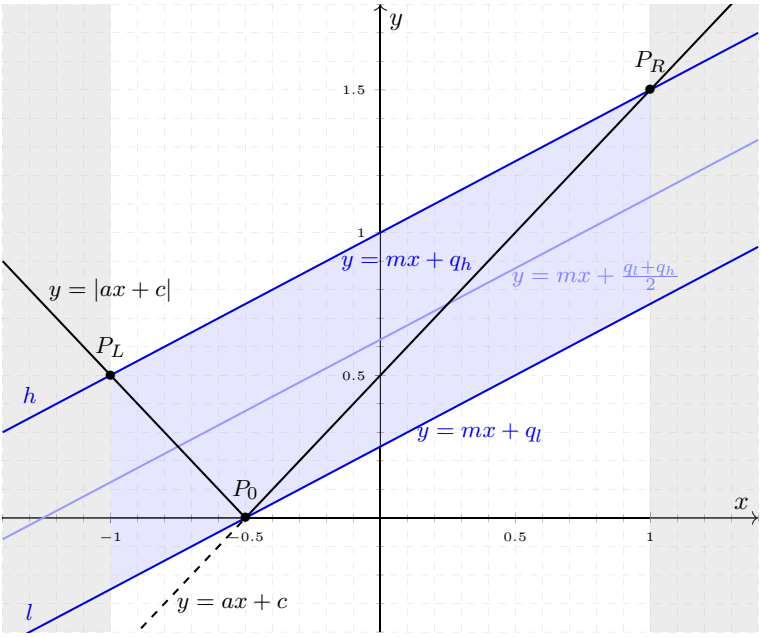

Fig. 2 Example of the absolute value of a zonotope

Given $\hat{a} = a_0 + \sum_{j=1}^{m} a_j \epsilon_j \in \mathcal{Z}$ and $\hat{b} = b_0 + \sum_{j=1}^{m} b_j \epsilon_j \in \mathcal{Z}$, the abstract operations are given below:

addition: $\hat{a} +^{\mathcal{Z}} \hat{b} \triangleq (a_0 + b_0) + \sum_{j=1}^{m} (a_j + b_j)\epsilon_j$

constant multiplication: $c\hat{a} \triangleq ca_0 + \sum_{j=1}^{k} ca_j \epsilon_j$

multiplication: $\hat{a} \cdot^{\mathcal{Z}} \hat{b} \triangleq \left(a_0 b_0 + \frac{1}{2} \sum_{j=1}^{m} |a_j b_j|\right) +$
$\sum_{j=1}^{m} (a_j b_0 + b_j a_0)\epsilon_j + \left(\frac{1}{2} \sum_{j=1}^{m} |a_j b_j| + \sum_{1 \le i < j \le m} |a_i b_j + a_j b_i|\right)\epsilon_{\mathrm{f}}$

exponential: $\hat{a}^{p^{\mathcal{Z}}} \triangleq \hat{a} \cdot^{\mathcal{Z}} \ldots \cdot^{\mathcal{Z}} \hat{a}$ with $p - 1$ abstract multiplications $\cdot^{\mathcal{Z}}$

dominance test: $\hat{a} <^{\mathcal{Z}} \hat{b} \triangleq a_0 - b_0 + \sum_{j=1}^{m} |a_j - b_j| < 0$

An abstract modulus on zonotopes To the best of our knowledge, no algorithm implementing a sound abstraction of the modulus operation on zonotopes is available in the literature. Therefore, we designed a novel abstract function on $\mathcal{Z}$ that approximates the generic modulus operation. By following the general approach in affine arithmetic described in [9], we define a zonotope approximating the absolute value of a given zonotope, and then, we compute the maximal absolute error of such approximation and add that error to a nonlinear term ϵ_{f} to guarantee soundness.

Figure 2 depicts an example where a zonotope $\hat{a} = c + a\epsilon_1$ is plotted as $y = ax + c$, with $x \in [-1, 1]$ (the white area in the diagram), through a dashed black line, and its absolute value $y = |\hat{a}| = |ax + c|$ as solid black line. In this example, finding a sound over-

approximation of $|\hat{a}|$ in $\mathcal{Z}$ means computing two parallel lines defining a zonotope which includes every point of $y = |\hat{a}|$, as shown in the figure by the blue area. The two lines l and h determining the blue area are parallel, i.e., they have the same slope $m \in \mathbb{R}$, and differ for their vertical displacements $q_l, q_h \in \mathbb{R}$. More precisely, we need to find $m, q_l, q_h \in \mathbb{R}$ such that $mx + q_l \leq |ax + c| \leq mx + q_h$ for every $x \in [-1, 1]$. The overapproximating zonotope will be generated by the line $y = mx + \frac{q_l+q_h}{2}$ parallel to l and h and will account for the absolute error $\frac{q_h-q_l}{2}$. This therefore defines the zonotope $|\hat{a}|^{\mathcal{Z}} \triangleq \frac{q_l+q_h}{2} + m\epsilon_1 + \frac{q_h-q_l}{2}\epsilon_f$ that retains some information about the linear contribution of ϵ_1 and introduces a nonlinear contribution in ϵ_f.

To compute m, q_l, q_u satisfying $mx + q_l \leq |ax + c| \leq mx + q_h$ for every $x \in [-1, 1]$, we first observe that the value of a zonotope $c + a\epsilon_1$ is either always positive, always negative, or it crosses $y = 0$ in some point $x_0 \in [-1, 1]$. The first two cases are trivial as the absolute value can be simply omitted, possibly after changing the sign of the zonotope, so we focus on the last case where x_0 is $-\frac{c}{a}$. The absolute value is therefore strictly decreasing in $[-1, x_0]$ and strictly increasing in $[x_0, 1]$ due to the linearity of its argument. We have that $P_0 = (x_0, 0)$ is the (global) minimum point while $P_L = (-1, |-a+c|)$ and $P_R = (1, |a+c|)$ are the two (local) maximal points. Thus, the following three inequalities for m, q_l, q_u must hold: $mx_0 + q_l \leq 0$, $|-a+c| \leq -m + q_h$, and $|a+c| \leq m + q_h$. An easy way to find m, q_h satisfying $|a + c| \leq m + q_h$ is to pick the line passing through P_L and P_R, whose slope is $m = \frac{|a+c|-|a-c|}{1-(-1)} = \frac{|a+c|-|a-c|}{2}$. The value of q_h can be estimated by imposing either P_L or P_R to be a solution of $y_P = mx_P + q_h$, a line referred to as Γ_h. Since the absolute value of a linear function is a convex shape, it can be crossed by Γ_h in at most two points, which are precisely P_L and P_R: As no other point can intersect Γ_h, and P_L, P_R are the extreme points within the domain, every other point must belong to the same half-space identified by Γ_h. Moreover, since P_L and P_R are maximal points, every other point must be dominated by Γ_h, thus proving that the line Γ_h is a sound upper bound. Lastly, we need to determine q_l. By using a similar argument, we define the line Γ_l as $y = mx + q_l$ such that $P_0 \in \Gamma_l$. Once again, due to convexity, Γ_l can cross the absolute value in at most two points, which is P_0 with multiplicity two, hence any other point must belong to the same half-space. Since P_0 is the global minimum, it turns out that Γ_l is dominated by every other point and, consequently, the line Γ_l is a sound lower bound. Since the vertical distance between the two lines Γ_h and Γ_l is $q_h - q_l$, we can consider the parallel line located halfway between them and as absolute error the semi distance $\frac{q_h-q_l}{2}$: This therefore defines as sound abstraction of $|\hat{a}|$ the zonotope $|\hat{a}|^{\mathcal{Z}} = \frac{q_l+q_h}{2} + m\epsilon_1 + \frac{q_h-q_l}{2}\epsilon_f$.

The example described above assumes a single nonzero linear noise contribution for ϵ_1 for the argument zonotope $\hat{a} = c + a\epsilon_1$. This same approximation technique can be applied to the case where the argument zonotope has a nonlinear noise contribution ϵ_r and no linear noise ϵ_i, i.e., $\hat{a} = c + a\epsilon_r$. While the computations remain the same, the coefficient m must be added to the nonlinear term, so that, in this case, we have that $|\hat{a}|^{\mathcal{Z}} = \frac{q_l+q_h}{2} + (|m| + \frac{q_h-q_l}{2})\epsilon_r$. In practice, it is always possible to consider ϵ_r as a fresh independent noise symbol while converting a zonotope to its geometrical representation.

When the argument zonotope has both a nonzero linear and nonlinear noise, i.e., $\hat{a} = c + a\epsilon_1 + b\epsilon_r$, or, more in general, there are d linear noise contributions ϵ_i and a nonlinear noise ϵ_r, i.e., $\hat{a} = a_0 + \sum_{j=1}^{d} a_j\epsilon_j + a_r\epsilon_r$, we can generalize this approximation technique as follows. We first convert the argument zonotope $\hat{a}$ into a hyperplane Π in $\mathbb{R}^{d+2}$ by interpreting ϵ_r as x_{d+1} (so that $a_{d+1} = a_r$), and adding a dimension x_{d+2} to represent the dependent variable, and we set the constraints $x_1, x_2, \ldots, x_{d+1} \in [-1, 1]$ (while $x_{d+2} \in \mathbb{R}$). We then define a subset $S \subseteq \Pi$ by selecting $d + 2$ points from Π whose values for every independent

variable x_i are either -1 or $+1$, while the value of the dependent variable x_{d+2} is accordingly computed, that is,

$$S = \left\{\left(a_1 e_1, \ldots, a_d e_d, a_{d+1} e_{d+1}, a_0 + \sum\nolimits_{j=1}^{d+1} a_j e_j\right) \in \Pi \mid \exists j.e_j = -1, \forall i \neq j.e_i = 1\right\} \cup \left\{\left(a_1, \ldots, a_d, a_{d+1}, a_0 + \sum\nolimits_{j=1}^{d+1} a_j\right)\right\}.$$

If all the dependent values x_{d+2} of every vector in S are nonnegative, then $\hat{a}$ is always nonnegative, so that its absolute value is itself, i.e., $|\hat{a}|^{\mathcal{Z}} \triangleq \hat{a}$. Similarly, if all the values x_{d+2} are nonpositive, then $\hat{a}$ is nonpositive, so that its absolute value can be obtained simply by $|\hat{a}|^{\mathcal{Z}} \triangleq -\hat{a}$. In both cases, the absolute value $|\hat{a}|^{\mathcal{Z}}$ is an exact abstraction with no loss of precision. Otherwise, there exist two vectors $\mathbf{x}$ and $\mathbf{y}$ in S with $\mathbf{x}_{d+2} < 0$ and $\mathbf{y}_{d+2} > 0$, meaning that Π has a nonempty intersection with the hyperplane Π_0 defined as $\mathbf{x}_{d+2} = 0$. We then define the subset $S' = \{(\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d, \mathbf{x}_{d+1}, |\mathbf{x}_{d+2}|) \mid \mathbf{x} \in S\}$; namely, we switch the negative signs of $\mathbf{x}_{d+2}$. By doing so, S' is a subset of the extremal points of the absolute value of $\hat{a}$. Next, we compute the hyperplane Π_h containing every point in S'. This step needs to compute the determinant of a $(d+2) \times (d+2)$ matrix, which is nonsingular by construction. Such hyperplane Π_h provides an upper bound for the absolute value of $\hat{a}$. Then, the lower bound hyperplane Π_l, parallel to Π_h, is computed, thus having the same coefficients of variables of Π_h and a different constant term. To do so, we compute $\Pi' = \Pi \cap \Pi_0$, thus defining a subspace in $\mathbb{R}^{d'}$ for some $d' < d+2$. We sample a single vector P_0 of Π', whose existence is guaranteed by construction, and we use that vector to estimate the constant term of Π_l, so that Π_l provides a lower bound for the absolute value of $\hat{a}$. Then, we consider the hyperplane Π_m parallel to both Π_l and Π_h and equally distant from them, i.e., $\Pi_m = \frac{\Pi_l + \Pi_h}{2}$, and convert Π_m to a zonotope in $\mathbb{R}^d$ by adding the vertical distance $\frac{\Pi_h - \Pi_l}{2}$ to the nonlinear noise term. Such abstraction is sound because the vectors in Π_h, resp. Π_l, dominate, resp. are dominated, by the absolute values of the points in the argument zonotope $\hat{a}$.

It turns out that our algorithm for the stability certification of kNN relies on an abstract modulus function on zonotopes that always has a specific form, and this can be exploited to enhance the efficiency of computing the modulus. In fact, our certification algorithm always applies an abstract modulus of type $|a_0 + a_j \epsilon_j|^{\mathcal{Z}}$, for some $j \in [1, m]$, that is, the modulus of a line on a plane with unknown ϵ_j. Hence, the abstract modulus computes the line including the two extremal points $(-1, a_0 - a_j)$ and $(+1, a_0 + a_j)$ as a correct upper bound for $|a_0 + a_j \epsilon_j|$, and the parallel line passing through the point $(-\frac{a_0}{a_j}, 0)$ as a correct lower bound. We then consider the line $y = px + q$ parallel to these two lines and at the same distance $d > 0$ from them. This allows us to define as abstract modulus $|a_0 + a_j \epsilon_j|^{\mathcal{Z}} \triangleq q + p\epsilon_j + d\epsilon_{\mathrm{f}}$, where ϵ_{f} is a fresh noise symbol.

2.2 *k*NN classifiers

Consider a ground truth dataset $D \subseteq X \times L$, where $X \subseteq \mathbb{R}^n$ is an input space and L is a set of classification labels, and a distance function $\delta : X \times X \to \mathbb{R}_{\geq 0}$. Given $k \in \mathbb{N}^* \triangleq \mathbb{N} \setminus \{0\}$, a kNN classifier is modeled as a total function $C_{\delta,k} \colon X \to \wp(L)$, which maps an input sample $\mathbf{x} \in X$ into a nonempty set of labels, by first selecting in D the k nearest samples to $\mathbf{x}$ according to δ, and then returning the set of their most frequent labels. Hence, an output set including more than one label means a tie vote, and this justifies why we consider sets of labels as codomain of classifiers.

2.3 Stability and robustness

A perturbation $P : X \to \wp(X)$ of an input sample $\mathbf{x} \in X$ is a variation of its feature values defining a potential adversarial region $P(\mathbf{x}) \in \wp(X)$. A very common instance [6] is given by perturbations for the maximum norm $\|\cdot\|_\infty$: Given $\mathbf{x} \in \mathbb{R}^n$ and a magnitude $\tau > 0$, the ℓ_∞-perturbation is $P^\tau_\infty(\mathbf{x}) \triangleq \{\mathbf{w} \in \mathbb{R}^n \mid \max(|\mathbf{w}_1 - \mathbf{x}_1|, \ldots, |\mathbf{w}_n - \mathbf{x}_n|) \leq \tau\}$, i.e., the ℓ_∞-ball of radius τ centered in $\mathbf{x}$. This perturbation can be exactly represented through intervals and zonotopes, that is, $P^\tau_\infty(\mathbf{x}) = \gamma^{\mathcal{T}^n}(\langle[\mathbf{x}_1 - \tau, \mathbf{x}_1 + \tau], \ldots, [\mathbf{x}_n - \tau, \mathbf{x}_n + \tau]\rangle) = \gamma^{\mathcal{Z}^n}(\langle \frac{\mathbf{x}_1}{2} + \tau\epsilon_1, \ldots, \frac{\mathbf{x}_n}{2} + \tau\epsilon_n\rangle)$.

A classifier $C : X \to \wp(L)$ is *accurate* on a ground truth input $(\mathbf{x}, l_\mathbf{x}) \in D$ when $C(\mathbf{x}) = \{l_\mathbf{x}\}$. Moreover, C is *stable* over a region $R \subseteq X$, when $\cup_{\mathbf{w} \in R} C(\mathbf{w}) = \{l\}$ holds, for some $l \in L$. Stability means that a classifier does not change its output on a region of similar inputs and is an orthogonal notion with respect to accuracy, as it does not require to know the ground truth labels. If a classifier C is both accurate on an input $(\mathbf{x}, l_\mathbf{x})$ and stable over a perturbation $P(\mathbf{x})$ of $\mathbf{x}$, then C is *robust* on input $(\mathbf{x}, l_\mathbf{x})$ for $P(\mathbf{x})$, i.e., for all $\mathbf{w} \in P(\mathbf{x})$, $C(\mathbf{w}) = \{l_\mathbf{x}\}$ holds. Accordingly, stability and robustness metrics for a classifier C on some test set $T \subseteq X \times L$ are defined as the percentage of test samples $\mathbf{x} \in T$ for which C is stable/robust over a perturbation $P(\mathbf{x})$:

$$\text{STAB}(C, T) \triangleq |\{(\mathbf{x}, l_\mathbf{x}) \in T \mid C \text{ stable on } P(\mathbf{x})\}|/|T|$$
$$\text{ROB}(C, T) \triangleq |\{(\mathbf{x}, l_\mathbf{x}) \in T \mid C \text{ robust on } (\mathbf{x}, l_\mathbf{x}) \text{ for } P(\mathbf{x})\}|/|T|$$

2.4 Individual fairness

Our method can be also applied to certify *individual fairness* [10] that intuitively encodes the principle that "two individuals who are similar with respect to a particular task should be classified similarly." The similarity relation on the input space X is expressed in terms of a distance δ and a threshold $\tau > 0$ by considering $S_{\delta,\tau} \triangleq \{(\mathbf{x}, \mathbf{y}) \in X \times X \mid \delta(\mathbf{x}, \mathbf{y}) \leq \tau\}$. The distance metric δ is specific to the fairness problem, where [10] studies the total variation or relative ℓ_∞ distances. Then, given an individual $\mathbf{x} \in X$, a classifier $C : X \to \wp(L)$ is *individually fair* on $\mathbf{x}$ with respect to $S_{\delta,\tau}$ when:

$$\forall \mathbf{y} \in X \cdot (\mathbf{x}, \mathbf{y}) \in S_{\delta,\tau} \Rightarrow C(\mathbf{x}) = C(\mathbf{y}).$$

Thus, individual fairness for $\mathbf{x}$ holds if and only if for all $\mathbf{y} \in P^\tau_\delta(\mathbf{x})$, $C(\mathbf{x}) = C(\mathbf{y})$, where $P^\tau_\delta : X \to \wp(X)$ is the perturbation defined as $P^\tau_\delta(\mathbf{x}) \triangleq \{\mathbf{y} \in X \mid \delta(\mathbf{x}, \mathbf{y}) \leq \tau\}$. Hence, by leveraging this simple observation, individual fairness boils down to stability, so that their metrics coincide.

3 Abstract verification of *k*NN

Given a classifier $C : X \to \wp(L)$, a *sound abstraction* of C on a numerical abstraction $\langle A, \gamma^A\rangle$ is an algorithm $C^A : A \to \wp(L)$, which is sound, i.e.,

$$\text{for all } a \in A, \cup_{\mathbf{x} \in \gamma^A(a)} C(\mathbf{x}) \subseteq C^A(a)$$

holds. Thus, soundness means that $C^A(a)$ over-approximates all the output labels of C on inputs abstractly represented by $a \in A$. If this over-approximation is indeed a singleton then

C is *provably stable* over the region $\gamma^A(a)$, i.e., this approach provides a formal *stability certification*.

Theorem 3.1 (Abstract stability certification) *Let C^A be a sound abstraction of C and assume that a region $R \subseteq X$ is over-approximated by some $a \in A$. If $|C^A(a)| = 1$ then C is stable over R.*

Proof By hypothesis, there exists a label $l \in L$ such that $C^A(a) = \{l\}$. By soundness of C^A, $\cup_{\mathbf{x} \in \gamma^A(a)} C(\mathbf{x}) \subseteq \{l\}$. Since, for all $\mathbf{x}$, $C(\mathbf{x}) \neq \varnothing$, we have that for all $\mathbf{x} \in \gamma^A(a)$, $C(\mathbf{x}) = \{l\}$. Since $R \subseteq \gamma^A(a)$, we obtain that for all $\mathbf{y} \in R$, $C(\mathbf{y}) = \{l\}$, namely C is stable over R. □

It is worth remarking that the converse of Theorem 3.1, in general, does not hold, meaning that this stability certification method can be incomplete. This incompleteness may depend on an input abstract value $a \in A$ which does not represent exactly the adversarial region R or by a loss of precision in the abstract computations of C^A. The former issue can be settled by leveraging abstract domains which are capable to represent exactly the perturbation model of interest, as it is the case of the interval and zonotope abstractions for ℓ_∞-perturbations.

3.1 Abstract distance

The kNN algorithm relies on a distance $\delta : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ for determining the k nearest vectors to a given input sample. Although kNN is parametric on δ, Minkowski distance is the standard choice: given $p \in \mathbb{N}^*$,

$$\delta_p(\mathbf{x}, \mathbf{y}) \triangleq \sqrt[p]{\sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|^p}.$$

In particular, the two most common instances are for $p = 1, 2$:

$$\text{Manhattan distance: } \mu(\mathbf{x}, \mathbf{y}) \triangleq \delta_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |\mathbf{x}_i - \mathbf{y}_i|$$
$$\text{Euclidean distance: } \eta(\mathbf{x}, \mathbf{y}) \triangleq \delta_2(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} (\mathbf{x}_i - \mathbf{y}_i)^2}$$

Observe that kNN relies on the distance for relative comparisons only, so we can safely discharge the p-th root $\sqrt[p]{\cdot}$ in δ_p to simplify the computations. A numerical abstract domain must therefore provide sound abstractions of the operations used for computing these distances, namely addition (i.e., subtraction), exponential and modulus. We recalled in Sect. 2.1 the definitions of the abstract operations on intervals and zonotopes. Let us remark the following points.

(1) We need a sound *abstract dominance relation* to be used for comparing abstract distances, i.e., an algorithm $(\cdot <^A \cdot) : A \times A \to \{\mathbf{true}, \mathbf{?}\}$ such that

$$\text{if } a_1 <^A a_2 = \mathbf{true} \text{ then for all } x \in \gamma^A(a_1) \text{ and } y \in \gamma^A(a_2),\ x < y \text{ holds.}$$

The dominance tests for intervals and zonotopes have been given in Sect. 2.1. It is worth noticing that the dominance relation $<^{\mathcal{I}}$ for intervals boils down to the so-called interval order [14], while the relation $<^{\mathcal{Z}}$ for zonotopes may exploit their relational information as encoded by shared noise symbols: e.g., a comparison between zonotopes such as $-2 + 2\epsilon_1 <^{\mathcal{Z}} 1 + \epsilon_1 + \epsilon_2$ reduces to $-3 + \epsilon_1 - \epsilon_2 <^{\mathcal{Z}} 0$, which clearly holds.

(2) A sound and precise enough approximation for zonotopes of the modulus function $|\cdot|$ was not available in the literature, and hence, we designed a novel algorithm for the abstract modulus of zonotopes as described in Sect. 2.1.2.

(3) The abstract operations on the product domains $\mathcal{I}^n$ and $\mathcal{Z}^n$ are defined by a straightforward componentwise extension of their unary versions on $\mathcal{I}$ and $\mathcal{Z}$.

It turns out that the abstract Minkowski distance $\delta_p^{\mathcal{I}^n}$, without the p-th root, on intervals does not lose precision, i.e., it is an exact approximation.

Theorem 3.2 (Minkowski distance on intervals is exact) *Given* $\mathbf{a}, \mathbf{b} \in \mathcal{I}^n$,

$$\{\delta_p(\mathbf{x}, \mathbf{y}) \mid \mathbf{x} \in \gamma^{\mathcal{I}^n}(\mathbf{a}), \mathbf{y} \in \gamma^{\mathcal{I}^n}(\mathbf{b})\} = \gamma^{\mathcal{I}}(\delta_p^{\mathcal{I}^n}(\mathbf{a}, \mathbf{b})),$$

where $\delta_p^{\mathcal{I}}(\mathbf{a}, \mathbf{b}) \triangleq (+^{\mathcal{I}})_{i=1}^{n}(|\mathbf{a}_i -^{\mathcal{I}} \mathbf{b}_i|^{\mathcal{I}})^{p^{\mathcal{I}}}$.

Proof We show that all the abstract operations on the interval abstract domain $\mathcal{I}$ used in the definition of $\delta_p^{\mathcal{I}}$ are complete. This entails the completeness of the interval Minkowski distance $\delta_p^{\mathcal{I}}$ because the composition of complete abstract functions preserves their completeness. In fact, if A is a numerical abstraction of $\wp(\mathbb{R}^n)$ (this property actually holds for generic domains in abstract interpretation) and $f^A : A^i \to A^j$ and $g^A : A^j \to A^k$ are two abstract functions that are complete w.r.t., resp., $f : \wp(\mathbb{R}^n)^i \to \wp(\mathbb{R}^n)^j$ and $g : \wp(\mathbb{R}^n)^j \to \wp(\mathbb{R}^n)^k$, then $g^A \circ f^A : A^i \to A^k$ is complete for $g \circ f : \wp(\mathbb{R}^n)^i \to \wp(\mathbb{R}^n)^k$ because $\gamma^A \circ g^A \circ f^A = g \circ \gamma^A \circ f^A = g \circ f \circ \gamma^A$.
We refer to the definitions of abstract numerical operations on the interval abstraction as given in Sect. 2.1.1. The interval difference $a_i -^{\mathcal{I}} b_i$ is well known to be complete (see, e.g., [31]). Then, we observe that the interval modulus is also complete, i.e., $\gamma^{\mathcal{I}}(|[l, u]|^{\mathcal{I}}) = \{|x| \in \mathbb{R} \mid l \leq x \leq u\}$: This is an easy observation which can be inferred by distinguishing the two cases $lu \geq 0$ and $lu < 0$ of the definition of $|[l, u]|^{\mathcal{I}}$. As a consequence, each interval $|\mathbf{a}_i -^{\mathcal{I}} \mathbf{b}_i|^{\mathcal{I}} = [l_i, u_i]$ occurring in the definition of $\delta_p^{\mathcal{I}}(\mathbf{a}, \mathbf{b})$ is such that $l_i \geq 0$. Therefore, by definition of interval exponential $(\cdot)^{p^{\mathcal{I}}}$, it turns out that $(|\mathbf{a}_i -^{\mathcal{I}} \mathbf{b}_i|^{\mathcal{I}})^{p^{\mathcal{I}}} = [l_i, u_i]^{p^{\mathcal{I}}} = [(l_i)^p, (u_i)^p] = \{x^p \in \mathbb{R} \mid l_i \leq x \leq u_i\}$; namely, completeness of the p-th interval exponential holds. Finally, interval addition is well known to be complete (see, e.g., [31]).
Hence, it turns out that the interval Minkowski distance $\delta_p^{\mathcal{I}}(\mathbf{a}, \mathbf{b})$ is a composition of complete abstract operations, so that $\delta_p^{\mathcal{I}}$ turns out to be complete. □

By contrast, we show that the abstract Minkowski distance on zonotopes cannot be guaranteed to be exact: This is expected as the modulus and exponential operations are not linear and, therefore, are necessarily approximated on zonotopes.

Example 3.3 (*Minkowski distance on zonotopes is not exact*) Consider two zonotopes $\hat{a} = 4 + \epsilon_1 + 2\epsilon_2$ and $\hat{b} = 2 + \epsilon_1 + \epsilon_2$, representing some feature in $\mathbb{R}$, that share two noise symbols ϵ_1, ϵ_2. Consider the abstract Euclidean distance $\eta^{\mathcal{Z}}(\hat{a}, \hat{b}) = (\hat{a} -^{\mathcal{Z}} \hat{b})^{2^{\mathcal{Z}}}$. Thus, by applying the operations on zonotopes recalled in Sect. 2.1.2, we have that:

$$\begin{aligned}\eta^{\mathcal{Z}}(\hat{a}, \hat{b}) &= \big((4 + \epsilon_1 + 2\epsilon_2) -^{\mathcal{Z}} (2 + \epsilon_1 + \epsilon_2)\big)^{2^{\mathcal{Z}}} = (2 + \epsilon_2)^{2^{\mathcal{Z}}} \\ &= (2 + \epsilon_2) \cdot^{\mathcal{Z}} (2 + \epsilon_2) = \tfrac{9}{2} + 4\epsilon_2 + \tfrac{1}{2}\epsilon_{\mathrm{f}}\end{aligned}$$

with $\epsilon_{\mathrm{f}} \in [0, 1]$ because this nonlinear noise symbol approximates a square which is always positive. (With $\epsilon_{\mathrm{f}} \in [-1, 1]$, the approximation would be even worse.) Thus, we have that $\gamma^{\mathcal{Z}}(\frac{9}{2} + 4\epsilon_2 + \frac{1}{2}\epsilon_{\mathrm{f}}) = [0.5, 9]$. However, observe that the square operation $(2 + \epsilon_2)^{2^{\mathcal{Z}}}$ is sound but not exact, because the range of values of $(2 + \epsilon_2)^2$ is the interval $[1, 3]^2 = [1, 9]$. Thus, $\{\eta(x, y) \in \mathbb{R} \mid x \in \gamma^{\mathcal{Z}}(\hat{a}), y \in \gamma^{\mathcal{Z}}(\hat{b})\} \subsetneq \gamma^{\mathcal{Z}}(\eta^{\mathcal{Z}}(\hat{a}, \hat{b}))$, as $[1, 9] \subsetneq [0.5, 9]$. □

Exactness of the distance function is not enough to achieve completeness of the abstract kNN classifier on intervals, as shown by the following example.

Example 3.4 (*Incompleteness of abstract kNN on intervals*) Let us consider a dataset $D = \{(\mathbf{v} = 2, l_1), (\mathbf{w} = 3, l_2)\}$ in the one-dimensional input space $\mathbb{R}$, and the 1NN classifier $C_{\mu,1}$ for the Manhattan distance μ. Consider a region $R = P^1_\infty(0) = \{\mathbf{x} \in \mathbb{R} \mid -1 \leq \mathbf{x} \leq 1\} \in \wp(\mathbb{R})$. The distances of a generic adversarial vector $\mathbf{x} \in R$ from $\mathbf{v}$ and $\mathbf{w}$ are:

$$\mu(\mathbf{x}, \mathbf{v}) = |\mathbf{x} - 2| = 2 - \mathbf{x},$$
$$\mu(\mathbf{x}, \mathbf{w}) = |\mathbf{x} - 3| = 3 - \mathbf{x}.$$

Hence, the dominance test $\mu(\mathbf{x}, \mathbf{v}) <^? \mu(\mathbf{x}, \mathbf{w})$ boils down to $2 - \mathbf{x} <^? 3 - \mathbf{x}$, which always holds. Thus, $\mathbf{v}$ is always the nearest neighbor to R, and, in turn, every sample in R is classified by $C_{\mu,1}$ as l_1, so that stability holds.

Let us perform the abstract stability certification on $\mathcal{I}$, where the region R is exactly represented by the interval $a \triangleq [-1, 1]$. The abstract Manhattan distances are as follows:

$$\mu^{\mathcal{I}}(a, \mathbf{v}) = |[-1, 1] -^{\mathcal{I}} 2|^{\mathcal{I}} = |[-3, -1]|^{\mathcal{I}} = [1, 3],$$
$$\mu^{\mathcal{I}}(a, \mathbf{w}) = |[-1, 1] -^{\mathcal{I}} 3]|^{\mathcal{I}} = |[-4, -2]|^{\mathcal{I}} = [2, 4].$$

These abstract distances do not allow us to infer the nearest vector to a because $\mu^{\mathcal{I}}(a, \mathbf{v}^{\mathcal{I}}) \not<^{\mathcal{I}} \mu^{\mathcal{I}}(a, \mathbf{w}^{\mathcal{I}})$ and $\mu^{\mathcal{I}}(a, \mathbf{w}^{\mathcal{I}}) \not<^{\mathcal{I}} \mu^{\mathcal{I}}(a, \mathbf{v}^{\mathcal{I}})$.
We can easily adapt this counterexample to show the incompleteness for different distance functions, such as the Euclidean distance. By a simple symbolic computation, we can infer that $\mathbf{v}$ is the nearest neighbor when $\mathbf{x} < 2.5$; hence, once again, every sample in R is labeled as l_1. However, by applying the abstract Euclidean distance, which is complete on $\mathcal{I}$, we obtain $\eta^{\mathcal{I}}(a, \mathbf{v}) = [1, 9]$ and $\eta^{\mathcal{I}}(a, \mathbf{w}) = [4, 16]$, so that we cannot infer the stability on R. This lack of precision is rooted in the interval abstraction that does not keep track of multiple occurrences of the same variable $\mathbf{x}$ in different abstract distances.
More refined relational abstractions such as octagons or even convex polyhedra [31] would also fail. For instance, with the convex polyhedra abstraction $\mathcal{P}$ we would still have an inconclusive comparison $\mu^{\mathcal{P}}(a, \mathbf{v}) = 1 \leq \mathbf{x} \leq 3 \not<^{\mathcal{P}} 2 \leq \mathbf{x} \leq 4 = \mu^{\mathcal{P}}(a, \mathbf{w})$. On a positive side, the relational information of the zonotope abstraction $\mathcal{Z}$ in this case allows us to prove stability. In fact, the zonotope $\hat{a} \triangleq 0 + \epsilon_1 \in \mathcal{Z}$ exactly represents the region R by keeping track of the dependence on $\mathbf{x}$ through the noise symbol ϵ_1, so we have that:

$$\mu^{\mathcal{Z}}(\hat{a}, \mathbf{v}) = |0 + \epsilon_1 \ -^{\mathcal{Z}} 2|^{\mathcal{Z}} = |-2 + \epsilon_1|^{\mathcal{Z}} = 2 + \epsilon_1,$$
$$\mu^{\mathcal{Z}}(\hat{a}, \mathbf{w}) = |0 + \epsilon_1 \ -^{\mathcal{Z}} 3|^{\mathcal{Z}} = |-3 + \epsilon_1|^{\mathcal{Z}} = 3 + \epsilon_1.$$

Thus, $\mu^{\mathcal{Z}}(\hat{a}, \mathbf{v}) <^{\mathcal{Z}} \mu^{\mathcal{Z}}(\hat{a}, \mathbf{w})$ iff $2 + \epsilon_1 <^{\mathcal{Z}} 3 + \epsilon_1$, which clearly holds. □

Example 3.4 exhibits a well-known issue of compositional computations in nonrelational abstractions (see [31]): For example, an expression such as $x - x$ with $x \in [0, 1]$ is compositionally evaluated in $\mathcal{I}$ as $[0, 1] -^{\mathcal{I}} [0, 1] = [-1, 1]$, thus causing a significant loss of precision w.r.t. its concrete value $[0, 0]$.

The following example shows that even if zonotopes are more precise than intervals, it may happen that intervals prove the stability of some input sample, whereas zonotopes fail.

Example 3.5 (*Intervals vs zonotopes for proving stability*) Consider the dataset $D = \{(\mathbf{v} = 0, \ell_1), (\mathbf{w} = 4.1, \ell_2)\}$, a region $R = \{\mathbf{x} \mid 0 \leq \mathbf{x} \leq 2\}$, and the 1NN classifier for the Euclidean distance η (w.l.o.g. we consider the square of η). The region R is exactly represented by the

interval $a = [0, 2] \in \mathcal{I}$ and by the zonotope $\hat{a} = 1 + \epsilon_1 \in \mathcal{Z}$. The abstract Euclidean distances on $\mathcal{I}$ and $\mathcal{Z}$ are as follows:

$$\eta^{\mathcal{I}}(a, \mathbf{v}) = ([0, 2] -^{\mathcal{I}} 0)^{2^{\mathcal{I}}} = [0, 4],$$
$$\eta^{\mathcal{I}}(a, \mathbf{w}) = ([0, 2] -^{\mathcal{I}} 4.1)^{2^{\mathcal{I}}} = [4.41, 16.81],$$
$$\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v}) = (1 + 1\epsilon_1 -^{\mathcal{Z}} 0)^{2^{\mathcal{Z}}} = 1 + 2\epsilon_1 + \epsilon_{f_1}, \quad \text{with } \epsilon_{f_1} \in [0, 1],$$
$$\eta^{\mathcal{Z}}(\hat{a}, \mathbf{w}) = (1 + 1\epsilon_1 -^{\mathcal{Z}} 4.1)^{2^{\mathcal{Z}}} = 9.61 - 6.2\epsilon_1 + \epsilon_{f_2}, \quad \text{with } \epsilon_{f_2} \in [0, 1].$$

Thus, for intervals, we have that $\eta^{\mathcal{I}}(a, \mathbf{v}) <^{\mathcal{I}} \eta^{\mathcal{I}}(a, \mathbf{w})$ iff $[0, 4] <^{\mathcal{I}} [4.41, 16.81]$, which holds and, therefore, entails stability. For zonotopes, we have that:

$$\begin{aligned} &\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v}) <^{\mathcal{Z}} \eta^{\mathcal{Z}}(\hat{a}, \mathbf{w}) && \text{iff} \\ &1 + 2\epsilon_1 + \epsilon_{f_1} <^{\mathcal{Z}} 9.61 - 6.2\epsilon_1 + \epsilon_{f_2} && \text{iff} \\ &-8.61 + 8.2\epsilon_1 + \epsilon_{f_1} - \epsilon_{f_2} <^{\mathcal{Z}} 0 \end{aligned}$$

which does not hold for, e.g., $\epsilon_1 = 1$, $\epsilon_{f_1} = 1$ and $\epsilon_{f_2} = 0$. Thus, stability cannot be proved with $\mathcal{Z}$. Let us remark that zonotopes here fail because $\mathcal{Z}$ needs to introduce two different fresh nonlinear noise symbols ϵ_{f_1} and ϵ_{f_2} for computing, resp., $\eta^{\mathcal{Z}}(\hat{a}, \mathbf{v})$ and $\eta^{\mathcal{Z}}(\hat{a}, \mathbf{w})$, while both would represent the same square ϵ_1^2. □

Example 3.5 arises because zonotopes do not keep track precisely of all nonlinear terms, as for the p-th Minkowski distance in $\mathbb{R}^n$ this would require storing and computing n^p nonlinear terms, thus making abstract computations for practical datasets unfeasible (see [19] for further details on the approximations and practical limitations of zonotopes).

3.2 Abstract *k*NN classification

Given a ground truth dataset D, we describe an algorithm for computing the sound abstract kNN classifier $C^A_{\delta,k}$ on a numerical abstract domain A, which is parametric on a distance function δ, provided that A is endowed with the abstract functions to be used for designing a sound abstract distance $\delta^A : A \times A \to A$, where, by a slight abuse of notation, A used in the domain $A \times A$ of δ^A is meant to be an abstraction of sets of vectors in $\wp(\mathbb{R}^n)$, while A used as codomain of δ^A is an abstraction of sets of numbers in $\wp(\mathbb{R})$; in this latter case, for each $a \in A$, we assume that $\mathrm{lb}(a), \mathrm{ub}(a) \in \mathbb{R} \cup \{-\infty, +\infty\}$ provide, resp., a sound lower and upper bound for $\gamma^A(a) \in \wp(\mathbb{R})$, i.e., for all $x \in \gamma^A(a)$, $\mathrm{lb}(a) \leq x \leq \mathrm{ub}(a)$ holds. The pseudocode for $C^A_{\delta,k}$ is given as Algorithm 1.

STEP$_1$: Computing and ordering abstract distances

Given a kNN classifier $C_{\delta,k}$, an input $(\mathbf{x}, l_\mathbf{x}) \in X \times L$, and a perturbation function $P : X \to \wp(X)$, we first need a sound abstraction $a_{P(\mathbf{x})} \in A$ for the region $P(\mathbf{x})$, and an abstract representation $\mathbf{y}^A \in A$ for every vector $\mathbf{y}$ occurring in the dataset as $(\mathbf{y}, l_\mathbf{y}) \in D$. For abstract domains that admit an abstraction function $\alpha^A : \wp(\mathbb{R}^n) \to A$, we define $a_{P(\mathbf{x})} \triangleq \alpha^A(P(\mathbf{x}))$. This can always be done for intervals where, for nonempty S, $\alpha^{\mathcal{I}}(S) \triangleq [\inf S, \sup S]$, whereas zonotopes, in general, do not admit an abstraction function. On the other hand, let us recall that both intervals and zonotopes provide exact abstract representations for ℓ_∞ perturbations $P^\epsilon_\infty(\mathbf{x})$. For each sample $(\mathbf{y}, l_\mathbf{y}) \in D$, we compute its abstract distance $d^A_\mathbf{y} \triangleq \delta^A(a_{P(\mathbf{x})}, \mathbf{y}^A) \in A$ from the abstract value $a_{P(\mathbf{x})}$ representing the perturbation $P(\mathbf{x})$.

Each abstract distance is paired with its corresponding label, thus constructing the set of pairs $\{(d^A_{\mathbf{y}}, l_{\mathbf{y}})\}_{(\mathbf{y},l_{\mathbf{y}})\in D}$. The abstract dominance relation $<^A$ on A is extended to $A \times L$ simply by disregarding labels, i.e., $(d^A_{\mathbf{y}}, l_{\mathbf{y}}) <^{A\times L} (d^A_{\mathbf{z}}, l_{\mathbf{z}})$ when $d^A_{\mathbf{y}} <^A d^A_{\mathbf{z}}$. This relation $<^{A\times L}$ is weakened by the following total order relation $\preceq$:

$$\left(d^A_{\mathbf{y}}, l_{\mathbf{y}}\right) \preceq \left(d^A_{\mathbf{z}}, l_{\mathbf{z}}\right) \stackrel{\triangle}{\Longleftrightarrow} \mathrm{lb}\left(d^A_{\mathbf{y}}\right) < \mathrm{lb}\left(d^A_{\mathbf{z}}\right) \text{ or } \left(\mathrm{lb}\left(d^A_{\mathbf{y}}\right) = \mathrm{lb}\left(d^A_{\mathbf{z}}\right) \,\&\, \mathrm{ub}\left(d^A_{\mathbf{y}}\right) \leq \mathrm{ub}(d^A_{\mathbf{z}})\right) \quad (*)$$

where $\prec$ denotes the corresponding strict order relation. This relation $(*)$ allows us to sort the set $\{(d^A_{\mathbf{y}}, l_{\mathbf{y}})\}_{(\mathbf{y},l_{\mathbf{y}})\in D}$ into a totally ordered set $\langle O, \preceq\rangle$. By a slight abuse of notation, we refer to $O[i]$, with $i \in [1, |D|]$, as the i-th smallest element of the total order $\langle O, \preceq\rangle$, so that $O[1]$ is the smallest element, $O[2]$ the second smallest, and so forth. Firstly, let us observe that $\preceq$ weakens $<^{A\times L}$, because if $O[i] <^{A\times L} O[j]$ holds, then it turns out that $\mathrm{lb}(O[i]) \leq \mathrm{ub}(O[i]) < \mathrm{lb}(O[j])$, so that $O[i] \prec O[j]$ holds, meaning that $i < j$. Moreover, a second property of the total order $\langle O, \preceq\rangle$ is that if $O[j]$ dominates $O[i]$, then any entry $O[k]$ with index $k \geq j$ also dominates $O[i]$, i.e., $O[i] <^{A\times L} O[j]$ implies $\forall k \geq j$, $O[i] <^{A\times L} O[k]$. In fact, $k > j$ implies $O[j] \preceq O[k]$, so that $\mathrm{lb}(O[j]) \leq \mathrm{lb}(O[k])$, and, in turn, since $O[i] <^{A\times L} O[j]$, we have that $\mathrm{ub}(O[i]) < \mathrm{lb}(O[j]) \leq \mathrm{lb}(O[k])$, hence entailing that $O[i] <^{A\times L} O[k]$ holds. In the best case scenario, the sequence $\langle O[i]\rangle_{i\in[1,|D|]}$ may result to be a total order for $<^{A\times L}$, meaning that for all $i, j \in [1, |D|]$, if $i < j$, then $O[i] <^{A\times L} O[j]$. In this optimal case, the abstract computation of the k nearest neighbors of $a_{P(\mathbf{x})}$ boils down to extracting the first k elements from the sequence O. However, in general, $\langle O[i]\rangle_{i\in[1,|D|]}$ will not be totally ordered for $<^{A\times L}$ because abstract distances may "overlap," as illustrated in Example 3.4 for the intervals $[1, 3]$ and $[2, 4]$. In our NAVe tool, O has been implemented as a min heap for the total order $\preceq$ (cf. MinHeapify$(O, \preceq)$ at line 6 of Algorithm 1) to leverage its logarithmic cost for building heaps and extracting its i-th smallest element.

STEP$_2$: Computing score bounds for labels

We compute the *abstract score intervals* $\mathsf{s}[l] \in \mathcal{I}$, for all the labels $l \in L$, namely an integer interval $\mathsf{s}[l] = [\mathrm{lb}(l), \mathrm{ub}(l)]$, with $\mathrm{lb}(l), \mathrm{ub}(l) \in \mathbb{N}$, that provides a lower bound $\mathrm{lb}(l) \geq 0$ and an upper bound $\mathrm{ub}(l) \geq \mathrm{lb}(l)$ to the number of votes that a label l receives from the k nearest neighbors of $a_{P(\mathbf{x})}$. We initialize $\mathsf{s}[l] = [0, 0]$, for each label $l \in L$, then we extract the first k pairs from the indexed sequence $\langle O[i]\rangle_{i\in[1,|D|]}$ of STEP$_1$. For each extracted pair $(d^A_{\mathbf{z}}, l_{\mathbf{z}})$, we check whether O still includes a pair $(d^A_{\mathbf{y}}, l_{\mathbf{y}})$ having a different label and not dominating $d^A_{\mathbf{z}}$, i.e., such that $l_{\mathbf{y}} \neq l_{\mathbf{z}}$ and $d^A_{\mathbf{z}} \not<^A d^A_{\mathbf{y}}$. If such pair does not exist, then all the pairs $(d^A_{\mathbf{y}}, l_{\mathbf{y}})$ left in O are such that $d^A_{\mathbf{z}} <^A d^A_{\mathbf{y}}$, thus meaning that $l_{\mathbf{z}}$ will certainly get a vote from $\mathbf{z}$, which has been proved to be a k-nearest neighbor of $a_{P(\mathbf{x})}$. If this happens then it is correct to increase by 1 both the lower and the upper bound of the interval of scores $\mathsf{s}[l_{\mathbf{z}}]$. Otherwise, it is not possible to infer that $l_{\mathbf{z}}$ will certainly get a vote from $\mathbf{z}$, so that the lower bound $\mathrm{lb}(l_{\mathbf{z}})$ cannot be increased, while to preserve the soundness of $\mathsf{s}[l_{\mathbf{z}}]$ we must increase its upper bound $\mathrm{ub}(l_{\mathbf{z}})$ by 1, meaning that it is possible that $l_{\mathbf{z}}$ will get an additional vote from $\mathbf{z}$. After this computation of the score intervals $[\mathrm{lb}(l), \mathrm{ub}(l)]_{l\in L}$ that processed the first k pairs extracted from the sequence O, the sum $\sigma_k \triangleq \sum_{l\in L} \mathrm{lb}(l)$ of the current lower bounds may be less than k, meaning that still no sound inference on the set of most voted labels for kNN can be drawn from the current status of the score intervals. Hence, if $\sigma_k < k$ and there exist unprocessed pairs $(d^A_{\mathbf{z}}, l_{\mathbf{z}})$ left in O whose distance $d^A_{\mathbf{z}}$ does not dominate all the distances of the first k pairs extracted from O, then we check whether $\mathrm{ub}(l_{\mathbf{z}}) < k - \sum_{l\in L\setminus\{l_{\mathbf{z}}\}} \mathrm{lb}(l)$ holds. If this is the case then $\mathrm{ub}(l_{\mathbf{z}})$ is increased by 1.

STEP$_3$: Refining lower bounds

Following STEP$_2$, we try to refine the lower bounds of $\mathsf{s}[l]$ as sketched by the following example. Let us consider a binary classification with two labels l_1 and l_2 and $k = 7$, whose current score intervals are, resp., $\mathsf{s}[l_1] = [2, 4]$ and $\mathsf{s}[l_2] = [1, 3]$. We observe that this information allows us to make a sound increment of the lower bounds of both l_1 and l_2. In fact, since the sum of the two labels must be $k = 7$, this can happen just when $\mathsf{s}[l_1] = [4, 4]$ and $\mathsf{s}[l_2] = [3, 3]$. Therefore, in this case, we can infer that l_1 is the most voted label.

A precise and general pseudocode of this refinement step is given at lines 17-19 of Algorithm 1. For each label l, we compute the minimum μ between k and the sum of $\mathrm{ub}(l')$ for all $l' \neq l$. If $k - \mu < \mathrm{lb}(l)$ holds then we can correctly refine the lower bound for l to $k - \mu$.

1: $M, O \leftarrow \varnothing$ ▷ M, O indexing starts at 1
2: **for all** $(\mathbf{y}, l_{\mathbf{y}}) \in D$ **do** ▷ STEP$_1$
3: $\quad \mathbf{y}^A \leftarrow \alpha(\mathbf{y})$
4: $\quad d^A_{\mathbf{y}} \leftarrow \delta^A(a_{P(\mathbf{x})}, \mathbf{y}^A)$
5: $\quad \mathrm{Insert}(O, (d^A_{\mathbf{y}}, l_{\mathbf{y}}))$
6: MakeTotalOrder($O, \preceq$) ▷ MinHeapify($O, \preceq$)
7: **for all** $l \in L$ **do** $\{\mathrm{lb}(l) \leftarrow 0; \mathrm{ub}(l) \leftarrow 0\}$ ▷ STEP$_2$
8: **for all** $i \in [1, k]$ **do** $M[i] \leftarrow \mathrm{Extract}(O[i])$
9: **for all** $(d^A_{\mathbf{z}}, l_{\mathbf{z}}) \in M$ **do**
10: $\quad \mathrm{ub}(l_{\mathbf{z}}) \leftarrow \mathrm{ub}(l_{\mathbf{z}}) + 1$
11: $\quad$ **if** $\forall (d^A_{\mathbf{y}}, l_{\mathbf{y}}) \in O,\ l_{\mathbf{y}} = l_{\mathbf{z}} \Rightarrow d^A_{\mathbf{z}} <^A d^A_{\mathbf{y}}$ **then** $\mathrm{lb}(l_{\mathbf{z}}) \leftarrow \mathrm{lb}(l_{\mathbf{z}}) + 1$
12: $\sigma_k \leftarrow \sum_{l \in L} \mathrm{lb}(l)$
13: **if** $\sigma_k < k$ **then**
14: $\quad$ **for all** $(d^A_{\mathbf{z}}, l_{\mathbf{z}}) \in O$ **do**
15: $\quad\quad$ **if** $\exists (d^A_{\mathbf{y}}, l_{\mathbf{y}}) \in M$ **such that** $l_{\mathbf{y}} \neq l_{\mathbf{z}} \wedge d^A_{\mathbf{y}} \not<^A d^A_{\mathbf{z}}$ **then**
16: $\quad\quad\quad$ **if** $\mathrm{ub}(l_{\mathbf{z}}) < k - (\sigma_k - \mathrm{lb}(l_{\mathbf{z}}))$ **then** $\mathrm{ub}(l_{\mathbf{z}}) \leftarrow \mathrm{ub}(l_{\mathbf{z}}) + 1$
17: **for all** $l \in L$ **do** ▷ STEP$_3$
18: $\quad \mu \leftarrow \min(k, \sum_{l' \neq l} \mathrm{ub}(l'))$
19: $\quad \mathrm{lb}(l) \leftarrow \max(\mathrm{lb}(l), k - \mu)$
20: **for all** $l \in L$ **do** ▷ STEP$_4$
21: $\quad$ **if** $\mathrm{ub}(l) = 0$ **then** $L \leftarrow L \setminus \{l\}$
22: **if** $(|L| = 1$ **or** $k = 1)$ **then return** L
23: $\tau \leftarrow \lceil \frac{k}{\min(k, |L|)} \rceil$
24: **return** $\{l \in L \mid \mathrm{ub}(l) \geq \tau,\ \forall l' \in L \setminus \{l\}.\ \mathsf{s}[l] \not<^{\mathcal{I}} \mathsf{s}[l']\}$

STEP$_4$: Abstract classification

After the refinement of STEP$_3$, we return the set of labels whose score intervals are numerically significant, i.e., different from $[0, 0]$, and maximal for the dominance relation $<^{\mathcal{I}}$ between score intervals, that is, $C^A_{\delta,k}(a_{P(\mathbf{x})})$ outputs the following set of labels:

$$\left\{ l \in L \mid \mathrm{ub}(l) \geq \left\lceil \frac{k}{\min(k, |L|)} \right\rceil,\ \forall l' \neq l \colon \mathsf{s}[l] \not<^{\mathcal{I}} \mathsf{s}[l'] \right\}.$$

We are thus excluding from the output set only those labels l whose score interval either has an upper bound strictly less than $\lceil \frac{k}{\min(k, |L|)} \rceil$ or is not maximal, i.e., there exists a different label l' with a dominant score $\mathsf{s}[l'] >^{\mathcal{I}} \mathsf{s}[l]$, meaning that the number of votes for l is surely less than the votes of l'. This definition is sound because no real classification label given as output by $C_{\delta,k}(\mathbf{y})$ for some adversarial attack $\mathbf{y} \in \gamma^A(a_{P(\mathbf{x})})$ is forgot, while the lack of precision in computing the abstract distances—this cannot happen with intervals but it may

be the case for zonotopes, cf. Theorem 3.2 and Example 3.3—and, in turn, the score intervals may lead to an over-approximation that includes some spurious labels.

Theorem 3.6 (Soundness of abstract kNN) *The abstract classifier $C^A_{\delta,k}$ is a sound approximation of $C_{\delta,k}$, namely for all $a \in A$, $\cup_{\mathbf{y} \in \gamma^A(a)} C_{\delta,k}(\mathbf{y}) \subseteq C^A_{\delta,k}(a)$.*

Proof It follows by the arguments given above that justify the soundness of the four steps of Algorithm 1 that implements the abstract classifier $C^A_{\delta,k}$. □

Remarks

In STEP$_1$, the first k pairs of the total order $\langle O, \preceq \rangle$ are intuitively the k most likely candidates to be the k nearest neighbors of the abstract adversarial region $a_{P(\mathbf{x})}$. If their distances from $a_{P(\mathbf{x})}$ are all strictly dominated by the other pairs in O then these first k samples in O are indeed the k nearest neighbors of $a_{P(\mathbf{x})}$, and therefore we can assign a sure vote to their labels, i.e., we increment both the lower and upper bounds of the score intervals for their labels. If, instead, this is not the case; namely, there exist $O[i] = (d_\mathbf{z}, l_\mathbf{z})$, for some $i \in [1, k]$, and $O[j] = (d_\mathbf{y}, l_\mathbf{y})$ with $j > k$, such that $d_\mathbf{z} \not\prec^A d_\mathbf{y}$, then we increment the upper bound $\text{ub}(l_\mathbf{z})$ just when $l_\mathbf{z} \neq l_\mathbf{y}$: In fact, if $l_\mathbf{z} = l_\mathbf{y}$, then neglecting the contribution of the sample $\mathbf{z}$ among the k nearest neighbors does not change the score for that same label $l_\mathbf{z}$. Moreover, if some $O[j] = (d_\mathbf{y}, l_\mathbf{y})$, with $j > k$, strictly dominates all the first k pairs of O, then all the pairs $O[m]$ with $m \geq k$ do the same, so that we do not need to consider them in computing the score intervals. The same reasoning applies to any pair $O[j] = (d_\mathbf{y}, l_\mathbf{y})$ w.r.t. a generic sample: If there exists some labeled sample $(\mathbf{u}, l_\mathbf{u})$ such that $l_\mathbf{u} \neq l_\mathbf{y}$ and $d_\mathbf{u} \not\prec^A d_\mathbf{y}$, then the upper bound $\text{ub}(l_\mathbf{y})$ can be correctly incremented by 1, as this label $l_\mathbf{y}$ could potentially be considered, although we do not know this for sure due to incompleteness. Increasing an upper bound of a score by some positive integer is always sound. However, while the computation of the abstract distance $\delta^A(a_{P(\mathbf{x})}, \mathbf{y}^A)$ may be exact (cf. Theorem 3.2), the computation of score intervals, in general, is not exact. This is due to the fact that score intervals for labels cannot represent relations between different scores. For example, mutual exclusion is a relational property which cannot be expressed by score intervals: The property "if a label $l_\mathbf{x}$ gets n votes, then a different label $l_\mathbf{y}$ gets $m - n$ votes" cannot be represented through intervals that cannot keep track of the fact that the score of $l_\mathbf{y}$ depends on that of $l_\mathbf{x}$.

3.3 Regression tasks

While our primary focus is on classification tasks, our methodology can be easily adapted to accommodate regression. Let us succinctly recall the basic steps of a regression task for kNN models. Initially, distances from a given input sample $\mathbf{x}$ to every point in the training dataset D are computed and exploited for sorting the vectors in D from nearest to farthest to $\mathbf{x}$, akin to the classification algorithm. Subsequently, the k nearest neighbors are identified, and an aggregation function is applied to their numeric values to compute the output regression value. A common example of aggregation function is the weighted mean, where the weights are inversely proportional to the distances.

Let us sketch how our abstract kNN algorithm can be adapted to a regression task. Firstly, we perform the same initial STEP$_1$ of the classification approach, thus computing and ordering abstract distances of training samples in D to an input abstract value $a \in A$. Following this, a sound superset of the k nearest neighbors can be inferred using analogous techniques as in classification (namely STEP$_2$), by leveraging an abstract version of the aggregation function.

The specific algorithm to be used for this purpose depends upon the chosen aggregation. Common aggregation functions, such as the weighted mean, entail using standard numerical operations such as addition, multiplication, and inverse, all of which have sound (or even exact) abstract versions on the abstract domains used in our work, notably intervals and zonotopes. Consequently, the abstract kNN regression algorithm should compute a sound output interval, namely a sound over-approximation of the true regression values for all the samples represented by input abstract value a.

3.4 Instantiating to different abstractions and perturbations

Our abstraction-based verification technique is fully parametric on the specific type of perturbation and numerical abstraction employed and is not restricted to interval-based perturbations/abstractions. As an example, let us sketch its applicability to perturbations not induced by the ℓ_∞ norm. Consider a perturbation function $P : \mathbb{R}^2 \to \wp(\mathbb{R}^2)$ defined by

$$P(\mathbf{x}) \triangleq \{\mathbf{x}' \in \mathbb{R}^2 \mid |\mathbf{x}_1 - \mathbf{x}'_1| + |\mathbf{x}_2 - \mathbf{x}'_2| \leq 1\}, \tag{‡}$$

thus representing the ℓ_1-ball of radius 1 with center $\mathbf{x}$. Geometrically, $P(\mathbf{x})$ is a square rotated by $\frac{\pi}{4}$ and cannot be exactly represented by an interval or a zonotope. Nevertheless, it is feasible to over-approximate $P(\mathbf{x})$ through the smallest two-dimensional interval (i.e., box) containing it simply by computing the minimum and maximum values for each axis, thus obtaining the box $\langle[\mathbf{x}_1 - 1, \mathbf{x}_1 + 1], [\mathbf{x}_2 - 1, \mathbf{x}_2 + 1]\rangle$. Indeed, each ℓ_p-ball of radius r can be over-approximated by a hypercube of the same radius, although this approximation may introduce a further loss of precision into the abstract kNN procedure that may yield an increased number of output labels, and, consequently, a less precise stability certification. To mitigate this, we can use more appropriate abstract domains that are capable of representing more precisely (or even exactly) a given class of perturbations, typically paying a cost in time efficiency: As an example, the octagon abstraction [30, 31] would allow to represent in an exact way the above perturbation (‡).

4 Equivalence of data poisoning and input perturbation for the maximum norm

Data poisoning is a distinctive form of attack that injects malicious or deceptive data into the training set of machine learning models [43, 47]. In contrast to conventional attacks that target vulnerabilities in model architecture or parameters, data poisoning surreptitiously erodes the fundamental underpinnings of the learning process since through subtle alterations or strategic injections of malicious instances into the training data, the attacker aims at compromising the integrity of the learning algorithm. We investigate the relationship between stability, akin to input poisoning, and data poisoning, by showing that our certification method for stability under input perturbations can be also applied to verify resilience to data poisoning when the underlying numerical abstract domain is the interval or zonotope abstraction.

Assume that training datasets D range into a space $\wp(X \times L)$. A data poisoning is defined as a function $\mathbb{P} : \wp(X \times L) \to \wp(X \times L)$ that for any input dataset D returns a poisoned dataset $\mathbb{P}(D)$. Consider a learning algorithm LA that, given a training dataset D, deterministically returns a classifier $\mathsf{LA}(D) = C_D : X \to \wp(L)$. A learning algorithm LA is defined to be resilient on an input sample $\mathbf{x} \in X$ under a data poisoning $\mathbb{P}$ when for all datasets D, $C_{\mathbb{P}(D)}(\mathbf{x}) = C_D(\mathbf{x})$. In the following, we consider data poisoning functions $\mathbb{P}^\tau_\infty$,

derived from maximum norm perturbations P^{τ}_{∞} and defined as follows:

$$\mathbb{P}^{\tau}_{\infty}(D) \triangleq \{(\mathbf{x}', l_{\mathbf{x}}) \in X \times L \mid (\mathbf{x}, l_{\mathbf{x}}) \in D,\ \mathbf{x}' \in P^{\tau}_{\infty}(\mathbf{x})\}.$$

We are interested in proving that a kNN classifier $C_{\delta,k}$ is resilient on some input to a $\mathbb{P}^{\tau}_{\infty}$ data poisoning of its ground truth dataset D. Since the output of an abstract classifier $C^{A}_{\delta,k}$ depends on the abstract distances of each sample $\mathbf{s}$ in the dataset D from the perturbation $P^{\tau}_{\infty}(\mathbf{x})$ of an input sample $\mathbf{x}$, in the following we prove that the abstract distance between an individual poisoning $P^{\tau}_{\infty}(\mathbf{s})$ of a sample $\mathbf{s}$ in D and a given input $\mathbf{x}$ coincides with the abstract distance between $\mathbf{s}$ and the corresponding perturbation $P^{\tau}_{\infty}(\mathbf{x})$ of $\mathbf{x}$. Hence, when this happens, by Theorems 3.1 and 3.6, it turns out that the resilience of a kNN classification $C_{\delta,k}(\mathbf{x})$ to a ℓ_{∞}-poisoning $\mathbb{P}^{\tau}_{\infty}$ of its training dataset D can be proved as an abstract stability certification of $C_{\delta,k}$ over $P^{\tau}_{\infty}(\mathbf{x})$ through the abstract classification $C^{A}_{\delta,k}(P^{\tau}_{\infty}(\mathbf{x}))$.

4.1 Intervals

Let us consider a training sample $\mathbf{s} \in D \subseteq \mathbb{R}^n$, a ℓ_{∞}-perturbation $P^{\tau}_{\infty} : \mathbb{R}^n \to \wp(\mathbb{R}^n)$, and an input sample $\mathbf{x} \in \mathbb{R}^n$. Hence, for the abstract Minkowski distance $\delta^{\mathcal{I}^n}_{p}$ that neglects the irrelevant p-th root, we have that:

$$\begin{aligned}
\delta^{\mathcal{I}^n}_{p}(\mathbf{x}, \alpha^{\mathcal{I}}(P^{\tau}_{\infty}(\mathbf{s}))) &= \sum\nolimits_{i=1}^{n} |\mathbf{x}_i -^{\mathcal{I}} [\mathbf{s}_i - \tau, \mathbf{s}_i + \tau]|^p \\
&= \sum\nolimits_{i=1}^{n} |[\mathbf{x}_i - \mathbf{s}_i - \tau, \mathbf{x}_i - \mathbf{s}_i + \tau]|^p \\
&= \sum\nolimits_{i=1}^{n} |[\mathbf{x}_i - \tau - \mathbf{s}_i, \mathbf{x}_i + \tau - \mathbf{s}_i]|^p \\
&= \sum\nolimits_{i=1}^{n} |[\mathbf{x}_i - \tau, \mathbf{x}_i + \tau] -^{\mathcal{I}} \mathbf{s}_i|^p \\
&= \delta^{\mathcal{I}^n}_{p}\big(\alpha^{\mathcal{I}}\big(P^{\tau}_{\infty}(\mathbf{x})\big), \mathbf{s}\big).
\end{aligned}$$

As a consequence, the resilience of a kNN classification $C_{\delta,k}(\mathbf{x})$ under a ℓ_{∞}-poisoning $\mathbb{P}^{\tau}_{\infty}$ of the training dataset D can be inferred as an abstract stability certification by means of the interval classification $C^{\mathcal{I}}_{\delta,k}(P^{\tau}_{\infty}(\mathbf{x}))$.

4.2 Zonotopes

A similar result can be proved for zonotopes. Let us recall that while, in general, the abstraction function for zonotopes does not exist, a ℓ_{∞}-perturbation $P^{\tau}_{\infty}(\mathbf{x})$ can be always exactly represented through the zonotope $\langle \frac{\mathbf{x}_1}{2} + \tau\epsilon_1, \ldots, \frac{\mathbf{x}_n}{2} + \tau\epsilon_n \rangle$, which is therefore the best abstraction of $P^{\tau}_{\infty}(\mathbf{x})$ in $\mathcal{Z}$. We have that:

$$\begin{aligned}
\delta^{\mathcal{Z}^n}_{p}\big(\mathbf{x}, \alpha^{\mathcal{Z}}\big(P^{\tau}_{\infty}(\mathbf{s})\big)\big) &= \sum\nolimits_{i=1}^{n} \left|\mathbf{x}_i - \left(\frac{\mathbf{s}_i}{2} + \tau\epsilon_i\right)\right|^p \\
&= \sum\nolimits_{i=1}^{n} \left|\mathbf{x}_i - \frac{\mathbf{s}_i}{2} - \tau\epsilon_i\right|^p \qquad [\text{as } \mathbf{x}_i - \tau\epsilon_i = \mathbf{x}_i + \tau\epsilon_i] \\
&= \sum\nolimits_{i=1}^{n} \left|\mathbf{x}_i + \tau\epsilon_i - \frac{\mathbf{s}_i}{2}\right|^p \\
&= \delta^{\mathcal{Z}^n}_{p}\big(\alpha^{\mathcal{Z}}\big(P^{\tau}_{\infty}(\mathbf{x})\big), \mathbf{s}\big).
\end{aligned}$$

Hence, resilience to a maximum norm data poisoning can be also inferred by leveraging the zonotope classifier $C^{\mathcal{Z}}_{\delta,k}$.

4.3 Arbitrary abstractions

In general, the equivalence shown above between maximum norm data poisoning and input perturbation does not hold for an arbitrary abstract domain A. To exhibit a counterexample, we consider an artificial abstraction $\mathcal{R}$ mirroring the behavior of the interval abstraction $\mathcal{I}$, with the exception that interval lower and upper bounds cannot belong to the range $(-1, 1)$. Thus, for example, a numerical set X having $\sup(X) \in (-1, 1)$ will have an interval approximation in $\mathcal{R}$ with upper bound 1 which is strictly larger than $\sup(X)$. Clearly, this abstract domain $\mathcal{R}$ is endowed with the abstraction function $\alpha^{\mathcal{R}}$, e.g., $\alpha^{\mathcal{R}}([-2, 0]) = [-2, 1]$.

Example 4.1 Let us consider the training dataset $D = \{\mathbf{s}_1 = ((1, 1), l_1),\ \mathbf{s}_2 = ((-1, -1), l_2),\ \mathbf{s}_3 = ((-2, -2), l_1)\}$ in $\mathbb{R}^2$, an input sample $\mathbf{x} = (3, 3)$, and the perturbation P^{τ}_{∞} with $\tau = 0.1$. The abstract Minkowski distances $d_i = \delta^{\mathcal{R}}_p(\alpha^{\mathcal{R}}(P^{\tau}_{\infty}(\mathbf{x})), \mathbf{s}_i)$ in $\mathcal{R}$ for the input perturbation are as follows:

$$
\begin{aligned}
d_1 &= [2(2-\tau)^p, 2(2+\tau)^p],\\
d_2 &= [2(4-\tau)^p, 2(4+\tau)^p],\\
d_3 &= [2(5-\tau)^p, 2(5+\tau)^p].
\end{aligned}
$$

Let us observe that $d_1 <^{\mathcal{R}} d_2 <^{\mathcal{R}} d_3$.
On the other hand, for the data poisoning $P^{\tau}_{\infty}(\mathbf{s}_i)$, we have the following abstract distances $e_i = \delta^{\mathcal{R}}_p(\mathbf{x}, \alpha^{\mathcal{R}}(P^{\tau}_{\infty}(\mathbf{s}_i)))$:

$$
\begin{aligned}
e_1 &= [2(2-\tau)^p, 2 \cdot 4^p],\\
e_2 &= [2 \cdot 2^p, 2(4+\tau)^p],\\
e_3 &= [2(5-\tau)^p, 2(5+\tau)^p].
\end{aligned}
$$

It turns out that both $e_1 <^{\mathcal{R}} e_3$ and $e_2 <^{\mathcal{R}} e_3$ hold but neither $e_1 <^{\mathcal{R}} e_2$ nor $e_2 <^{\mathcal{R}} e_1$ hold, since the abstract distances e_1 and e_2 overlap.
Hence, for the case $k = 1$, the abstract classifier $C^{\mathcal{R}}_{\delta_p,1}$ allows us to infer stability of input perturbation but not resilience to data poisoning. Stability of input perturbation can be inferred because the sample $\mathbf{s}_1$ is proved to be the nearest to $P^{\tau}_{\infty}(\mathbf{x})$. For resilience to data poisoning $P^{\tau}_{\infty}(\mathbf{s}_i)$, both $\mathbf{s}_1$ and $\mathbf{s}_2$ are selected by the abstract classifier $C^{\mathcal{R}}_{\delta_p,1}$, because the corresponding abstract distances to $\mathbf{x}$ overlap.

Nevertheless, we put forward some sufficient conditions on an abstraction A guaranteeing the equivalence of the best correct approximations in A of the distances for data poisoning and input perturbation for the maximum norm.

Theorem 4.2 (Equivalence of data poisoning and input perturbation) *Let $P^{\tau}_{\infty} : \mathbb{R}^n \to \wp(\mathbb{R}^n)$ be a ℓ_∞ perturbation and A be a numerical abstraction. Assume the following conditions:*

(i) *A admits an abstraction function α^A.*
(ii) *For all $\mathbf{x} \in X$, there exists $a_{\mathbf{x}}$ such that $P^{\tau}_{\infty}(\mathbf{x}) = \gamma^A(a_{\mathbf{x}})$, i.e., each adversarial region $P^{\tau}_{\infty}(\mathbf{x})$ is exactly representable in A.*
(iii) *For all $\mathbf{x} \in X$ and $\mathbf{t} \in \mathbb{R}^n$, $\mathbf{x} + \mathbf{t} \in P^{\tau}_{\infty}(\mathbf{x}) \Leftrightarrow \|\mathbf{t}\|_\infty \leq \tau$.*

Then, the best correct approximations in A of the distances for P^{τ}_{∞} input perturbation and data poisoning coincide.

Proof Since A admits an abstraction function α^A, the best correct approximation of any concrete function f exists and is $\alpha^A \circ f \circ \gamma^A$. Thus, given an input $\mathbf{x}$ and a training sample $\mathbf{s}$, to prove the equivalence we show that $\alpha^A(\delta_p(\gamma^A(\alpha^A(P^\tau_\infty(\mathbf{x}))), \mathbf{s})) = \alpha^A(\delta_p(\mathbf{x}, \gamma^A(\alpha^A(P^\tau_\infty(\mathbf{s})))))$. By letting $a \triangleq \alpha^A(P^\tau_\infty(\mathbf{x}))$ and $b \triangleq \alpha^A(P^\tau_\infty(\mathbf{s}))$, we show that $\delta_p(\gamma^A(a), \mathbf{s}) = \delta_p(\mathbf{x}, \gamma^A(b))$:

$$
\begin{aligned}
\delta_p(\gamma^A(a), \mathbf{s}) &= \bigcup\nolimits_{\mathbf{x}' \in \gamma^A(a)} \delta_p(\mathbf{x}', \mathbf{s}) && \text{[by (ii)]}\\
&= \bigcup\nolimits_{\mathbf{x}' \in P^\tau_\infty(\mathbf{x})} \delta_p(\mathbf{x}', \mathbf{s}) && \text{[by (iii)]}\\
&= \bigcup\nolimits_{\mathbf{t} \in \mathbb{R}^n, \|\mathbf{t}\|_\infty \leq \tau} \delta_p(\mathbf{x} + \mathbf{t}, \mathbf{s})\\
&= \bigcup\nolimits_{\mathbf{t} \in \mathbb{R}^n, \|\mathbf{t}\|_\infty \leq \tau} \sum\nolimits_{i=1}^n |\mathbf{x}_i + \mathbf{t}_i - \mathbf{s}_i|^p\\
&= \bigcup\nolimits_{\mathbf{t} \in \mathbb{R}^n, \|\mathbf{t}\|_\infty \leq \tau} \sum\nolimits_{i=1}^n |\mathbf{x}_i - (\mathbf{s}_i - \mathbf{t}_i)|^p\\
&= \bigcup\nolimits_{\mathbf{t} \in \mathbb{R}^n, \|\mathbf{t}\|_\infty \leq \tau} \delta_p(\mathbf{x}, \mathbf{s} - \mathbf{t}) && \text{[by def. of } \|\cdot\|_\infty\text{]}\\
&= \bigcup\nolimits_{\mathbf{t} \in \mathbb{R}^n, \|\mathbf{t}\|_\infty \leq \tau} \delta_p(\mathbf{x}, \mathbf{s} + \mathbf{t}) && \text{[by (iii)]}\\
&= \bigcup\nolimits_{\mathbf{s}' \in P^\tau_\infty(\mathbf{s})} \delta_p(\mathbf{x}, \mathbf{s}') && \text{[by (ii)]}\\
&= \bigcup\nolimits_{\mathbf{s}' \in \gamma^A(b)} \delta_p(\mathbf{x}, \mathbf{s}') = \delta_p(\mathbf{x}, \gamma^A(b)).
\end{aligned}
$$

□

Let us stress that Theorem 4.2 concerns abstract domains A endowed with an abstraction function and refers to the best correct approximations of distances, which may not coincide with the compositional definition of abstract distances considered in Sect. 3.1.

5 Dealing with categorical features

Datasets may contain both numerical and categorical features, where the latter usually range in nonnumerical sets of values, e.g., $color \in \{red, green, blue\}$. Most ML algorithms can only process numerical features, hence they rely on some numerical encoding of categorical features. *One-hot encoding* is a de facto standard encoding that consists in replacing a feature having k categories with k binary numerical features. More precisely, if $F = \{c_1, c_2, \ldots, c_q\}$ is the set of values for a categorical feature $f \in F$, one-hot encoding replaces f with q binary numerical features $(x^f_1, x^f_2, \ldots, x^f_q) \in \{0, 1\}^q$ in such a way that $\forall i \in [1, q]$: $x^f_i = 1 \Leftrightarrow f = c_i$. Therefore, one-hot encoding implicitly introduces the constraint $\sum_{i=1}^q x^f_i = 1$, which prevents a one-hot encoded sample from having more than one categorical value. If these relational constraints between one-hot encoded numerical features cannot be represented by an abstraction A, then an abstract classifier defined on A may exhibit a significant loss of precision, as illustrated by the following example for intervals.

Example 5.1 (*Loss of precision due to one-hot encoding*) Consider data samples with a categorical $color \in \{red, green, blue\}$ and a numerical $size \in \mathbb{R}_{\geq 0}$. Let $\mathbf{a}' \triangleq (red, 1)$, $\mathbf{b}' \triangleq (red, 3)$, and consider a dataset $D = \{(\mathbf{a}', l_1), (\mathbf{b}', l_2)\}$. By one-hot encoding, $color$ is replaced by $(isRed, isGreen, isBlue) \in \{0, 1\}^3$, so that $\mathbf{a}'$ and $\mathbf{b}'$ are encoded as $\mathbf{a} \triangleq (1, 0, 0, 1)$ and $\mathbf{b} \triangleq (1, 0, 0, 3)$. Consider an adversarial region $R \triangleq \{(r, g, b, size) \mid r, g, b \in$

$\{0, 1\}, size \in [0, 1]\}$. We observe that $\mathbf{a}$ is always closer than $\mathbf{b}$ to any vector $\mathbf{x} \in R$, for any Minkowski distance δ_p: In fact, we have that

$$\begin{aligned}\delta_p(\mathbf{a}, \mathbf{x}) < \delta_p(\mathbf{b}, \mathbf{x}) \Leftrightarrow & \sqrt[p]{|1 - \mathbf{x}_1|^p + \mathbf{x}_2^p + \mathbf{x}_3^p + |1 - \mathbf{x}_4|^p} < \\ & \sqrt[p]{|1 - \mathbf{x}_1|^p + \mathbf{x}_2^p + \mathbf{x}_3^p + |3 - \mathbf{x}_4|^p} \\ \Leftrightarrow & \ |1 - \mathbf{x}_4| < |3 - \mathbf{x}_4|\end{aligned}$$

which always holds for $\mathbf{x}_4 = \mathbf{x}_{size} \in [0, 1]$. Hence, 1NN classifies any vector in R as l_1.

Consider the abstract 1NN classifier on the interval abstraction $\mathcal{I}$ and the Manhattan distance δ_1. Therefore, R is abstracted as $\alpha^{\mathcal{I}^4}(R) = r = \langle [0, 1], [0, 1], [0, 1], [0, 1] \rangle \in \mathcal{I}^4$, and the abstract distances are as follows:

$$\begin{aligned}\delta_1^{\mathcal{I}}(r, \mathbf{a}) &= [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] = [0, 4], \\ \delta_1^{\mathcal{I}}(r, \mathbf{b}) &= [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [0, 1] +^{\mathcal{I}} [2, 3] = [2, 6].\end{aligned}$$

Since the intervals $[0, 4]$ and $[2, 6]$ overlap, we cannot infer which of the two samples $\mathbf{a}$ and $\mathbf{b}$ is the nearest to R, so that the abstract 1NN classifier returns $\{l_1, l_2\}$, i.e., no information at all. This loss of precision depends on the interval abstraction, which is not able to represent the constraint $isRed, isGreen, isBlue \in \{0, 1\}$ and $isRed + isGreen + isBlue = 1$. □

This additional loss of precision due to one-hot encoding could happen for zonotopes as well, although this phenomenon is mitigated by the chance that zonotopes represent some relational information between different one-hot encoded features through shared noise symbols.

To avoid the loss of precision due to one-hot encoding, we partition the original adversarial region R, abstractly represented by some $a \in A$, into q subregions $R_i \subseteq R$, each of them abstractly represented by some $a_i \in A$, where q is the overall number of values of the categorical features perturbed in the adversarial region R. Then, we execute the abstract classifier $C^A(a_i)$ for each abstract subregion a_i, and for each of them we compute a sound output set of labels. If, by repeatedly applying $C^A(a_i)$, it happens that the union of their output sets of labels is the whole set L, then we stop and output L. This splitting process will be such that every categorical feature of every subregion R_i will have exactly one possible categorical value, so that within each subregion R_i there is no need for abstracting the one-hot encoded categorical features. The final output will be obtained by collecting all the labels for each a_i, namely: $C^A(a) \triangleq \cup_{i \in [1,q]} C^A(a_i)$. This simple splitting strategy over categorical features reduces false negatives generated by one-hot encoding at the price of a higher certification time, since this procedure generates a new sub-problem for every possible combination of categorical values. Let us remark that if the perturbation of an input sample concerns categorical values only (i.e., numerical values are not perturbated)—this can happen in individual fairness certification—then this partitioning approach boils down to a concrete (and, therefore, trivially exact) verification, at the cost of an exponential number of sub-problems. More precisely, if m is the maximum number of different categories and p is the number of perturbed categorical features, then we need to check $O(m^p)$ sub-problems. This exponential blow-up is expected for an exact stability certification procedure with no false negatives. To balance cost and precision, one could allow only certain features to be split. (Unsplit features behave as numerical ones, and soundness still holds.) We applied this splitting technique in our experiments on individual fairness certification, where reference datasets typically include categorical features.

6 Experimental evaluation

We implemented our abstraction framework for kNN classifiers in a verification tool called NAVe and written in Python, and we instantiated it with the interval and zonotope abstractions. The source code of NAVe together with datasets and scripts for reproducing our experimental results is available on GitHub [12].

6.1 Setup

For our experiments, we considered some standard datasets used in robustness certification of kNN [42] and fairness verification of deep neural networks [29, 38]. Following [38], the datasets are preprocessed as follows:

(1) rows/columns with missing values are dropped;
(2) when needed (Letter, Pendigits and Satimage already have explicit test sets), datasets are split into training ($\approx$ 70–80%) and test ($\approx$ 20–30%) sets, resp., D and T;
(3) categorical features are one-hot encoded;
(4) numerical features are scaled to $[0, 1]$.

The details of these datasets, together with the accuracy of kNN on their test sets, are summarized in Table 1. In our individual fairness experiments, we consider the Noise-Cat similarity relation as defined by Ruoss et al. [38], where two individuals $\mathbf{x}, \mathbf{y} \in X$ are similar when:

(1) given the subset Noise $\subseteq \mathbb{N}$ of indexes of all numerical features and a noise threshold $\epsilon \geq 0$, for all $i \in$ Noise, $|\mathbf{x}_i - \mathbf{y}_i| \leq \epsilon$;
(2) given a subset Cat $\subseteq \mathbb{N}$ of indexes of "sensitive" categorical features, both $\mathbf{x}$ and $\mathbf{y}$ are allowed to have any category for features with indexes in Cat;
(3) every other categorical feature of $\mathbf{x}$ and $\mathbf{y}$, i.e., with index not in Cat, must be the same; namely, for any index $i \notin$ (Noise $\cup$ Cat), $\mathbf{x}_i = \mathbf{y}_i$ holds.

Fairness experiments with $\epsilon = 0$ represent a pure Cat perturbation of sensitive categorical features only, leaving numerical features unaltered: In this case, our certification method is complete, i.e., the percentages of individual fairness for $\epsilon = 0$ turn out to be exact (i.e., not a lower bound).

We instantiated our parametric abstract kNN classifier of Theorem 3.6 to both intervals $\mathcal{I}$ and zonotopes $\mathcal{Z}$, and we evaluated both the Manhattan δ_1 and Euclidean δ_2 distances. We considered the ℓ_∞-perturbation P^ϵ_∞ for our stability experiments, with the magnitude ϵ ranging in $[0.001, 0.1]$ for stability experiments ($[0.001, 0.05]$ for the dataset Letter), i.e., numerical features can be altered from $\pm 0.1\%$ to $\pm 10\%$. In the individual fairness experiments, we considered the following Noise-Cat perturbations: For Noise, the numerical attributes were perturbed with P^ϵ_∞ with $\epsilon \in [0, 0.05]$; for Cat, the sensitive categorical attributes were *race* for Compas and *gender* for German; when $\epsilon = 0$, this boils down to a pure Cat perturbation. The parameter k ranges in $\{1, 3, 5, 7\}$, where, following the standard practice for kNN, we avoided even values of k as they are more likely to introduce tie votes in the classification. We conducted all our experiments on a low-cost AWS virtual machine *t2.micro* instance, that provides a baseline level of CPU performance through a single 2.5 GHz CPU and 1 GB of RAM. Throughout the experiments, we mostly observed consistent time behaviors.

Table 1 Summary of datasets

Dataset	$\|D\|$ training	$\|T\|$ test	#feat	#feat with one-hot	#labels	kNN accuracy %			
						$k=1$	$k=3$	$k=5$	$k=7$
STABILITY									
Australian	483	207	14	39	2	77.8	80.2	82.6	82.6
BreastCancer	479	204	10	10	2	92.6	94.6	93.6	93.6
Diabetes	556	230	8	8	2	70.9	72.2	70.0	71.3
Fourclass	604	258	2	2	2	100	100	100	100
Letter	15,000	5000	16	16	26	95.7	94.6	94.2	94.3
Pendigits	7494	3498	16	16	10	97.7	97.8	97.5	97.5
Satimage	4435	2000	36	46	6	88.8	90.3	89.5	90.1
FAIRNESS									
Compas	4222	1056	10	370	2	58.4	59.1	60.2	61.1
German	800	200	20	56	2	73.0	71.5	74.5	77.0

6.2 Results

Tables 2, 3, 4, 5 report the percentages of test samples in T for which our NAVe tool proves that the kNN classifier is stable, i.e., for all k and ϵ, we provide the following metric:

$$\text{ProvableStability}_{k,\epsilon} \triangleq |\{(\mathbf{x}, \_) \in T \mid |C^{A}_{\delta_i,k}(P^{\epsilon}_{\infty}(\mathbf{x}))| = 1\}|/|T|$$

where $A \in \{\mathcal{I}, \mathcal{Z}\}$ and $i = 1, 2$. As shown in Sect. 2.4, for fairness datasets provable stability means provable individual fairness. For each distance δ_1 and δ_2, and for each dataset and perturbation magnitude ϵ, we highlight in bold the percentage corresponding to the most provably stable/fair kNN classifier. Due to incompleteness of the abstract kNN classification (cf. Example 3.4), it is worth recalling that $\text{ProvableStability}_{k,\epsilon}$ is a lower bound of the real stability of kNN on the test set T.

As expected, the zonotope abstraction $\mathcal{Z}$ allows us to have a certification technique that is generally more precise, and often much more precise, than that using the interval domain $\mathcal{I}$. The only exception is provided by the German dataset with $\epsilon = 0.02$ where for the case $k = 1$ intervals infer one more stable sample than zonotopes (overall, 85% vs. 84.5% of provable stability; indeed, this may happen as shown in Example 3.5).

Our NAVe tool infers with the zonotope abstraction more than 80% of stability, independently of k and distance δ_i, for:

(i) Australian for all $\epsilon \leq 0.1$;
(ii) BreastCancer for all $\epsilon \leq 0.05$;
(iii) Fourclass and Pendigits for all $\epsilon \leq 0.03$;
(iv) Diabetes, Letter and Satimage for $\epsilon \leq 0.005$.

Of course, provable stability decreases with higher values of ϵ since stronger perturbations are more likely to produce unstable behaviors, as well as more false positives among the approximate output sets of labels. In particular, we observe that Diabetes exhibits the worst stability scores, that together with a low accuracy ($\approx 70\%$) hints that a diagnosis of diabetes may be a hard task for which kNN does not perform well. On the other hand, the provable stability of Letter seems to be negatively affected by the size of its training set D, as more samples and more features are more likely to introduce ties between abstract distances.

Table 2 Percentages of provable stability or individual fairness for Intervals $\mathcal{I}$ with Manhattan distance δ_1 on the whole test sets T

ϵ	Australian				BreastCancer				Diabetes				Fourclass				Letter			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	98.5	**99.5**	99.0	99.0	**100**	98.5	99.5	**100**	93.4	94.3	93.4	**95.2**	**100**	99.6	**100**	**100**	**96.9**	94.4	93.4	93.5
0.005	**96.6**	96.1	93.2	95.1	97.5	**98.5**	**98.5**	97.5	**73.4**	69.1	63.9	65.2	**99.6**	**99.6**	99.2	99.2	**90.5**	86.7	83.6	81.2
0.01	92.2	93.2	91.3	**94.2**	93.6	95.5	**97.0**	96.5	**45.2**	39.1	36.9	34.3	**99.2**	98.8	96.1	95.7	**75.3**	67.2	60.8	56.1
0.02	88.8	88.4	86.9	**89.8**	85.7	86.2	86.7	**88.7**	**14.3**	12.6	10.8	9.1	**87.6**	86.0	81.7	81.0	**40.2**	32.6	27.9	25.0
0.03	84.0	83.0	**85.9**	85.5	78.9	83.3	84.8	**86.2**	**4.7**	3.0	1.7	1.3	**70.1**	68.6	67.0	62.4	**15.4**	12.0	9.84	8.60
0.05	79.7	80.1	**83.5**	82.6	66.6	68.6	**75.4**	75.0	**0.8**	**0.8**	0	0	34.1	**34.5**	29.8	28.6	**1.2**	1.1	1.1	1.0
0.07	**78.2**	**78.2**	77.7	**78.2**	34.8	45.1	51.4	**58.8**	**0.4**	0	0	0	15.1	**15.5**	14.3	13.5	–	–	–	–
0.10	**69.0**	64.2	65.7	66.6	5.3	5.8	6.3	**19.1**	0	0	0	0	**5.4**	5.0	5.0	4.2	–	–	–	–

ϵ	Pendigits				Satimage				ϵ	Compas				German			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	k = 3	k = 5	$k=7$		$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**99.5**	99.1	99.2	99.1	**93.6**	91.8	92.0	91.7	0	56.4	62.0	68.0	**70.5**	**85.5**	83.5	82.5	85.0
0.005	**96.5**	96.1	95.8	95.4	**68.3**	65.5	64.2	63.3	0.001	46.5	52.3	57.7	**60.0**	**84.5**	83.5	81.5	**84.5**
0.01	**92.1**	91.8	91.1	90.7	**44.4**	43.5	43.7	43.9	0.002	40.5	45.9	49.3	**51.7**	**84.0**	82.5	81.0	**84.0**
0.02	**79.0**	78.4	77.9	77.5	**20.8**	19.9	20.0	20.4	0.005	26.3	31.8	33.7	**36.3**	**83.5**	79.5	80.5	82.0
0.03	62.6	**63.6**	63.2	63.0	12.1	12.1	11.8	**12.2**	0.01	16.9	20.8	22.7	**26.0**	**82.0**	76.0	78.0	80.0
0.05	28.3	**29.5**	29.2	28.6	**8.4**	**8.4**	**8.4**	**8.4**	0.02	11.0	13.7	14.5	**16.9**	**78.0**	74.0	74.0	73.0
0.07	8.1	8.9	**9.1**	**9.1**	6.3	**6.4**	6.2	**6.4**	0.03	9.1	10.6	11.5	**13.6**	**73.5**	69.5	70.0	68.0
0.10	**0.2**	0.1	0.06	0.03	2.9	**3.0**	2.9	2.9	0.05	5.8	7.0	7.4	**8.9**	**67.5**	62.5	59.5	58.0

Table 3 Percentages of provable stability or individual fairness for Zonotopes $\mathcal{Z}$ with Manhattan distance δ_1 on the whole test sets T

ϵ	Australian				BreastCancer				Diabetes				Fourclass				Letter			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	99.5	99.5	99.5	**100**	99.0	99.5	**100**	95.2	96.5	96.5	96.0	**100**	99.6	**100**	**100**	**96.9**	94.4	93.4	93.5
0.005	**98.5**	97.5	94.6	95.6	98.0	98.5	99.0	**99.5**	**85.2**	82.1	83.4	83.9	99.6	**100**	**100**	99.6	**92.3**	89.7	88.2	87.7
0.01	**94.6**	**94.6**	92.7	**94.6**	96.0	96.5	**98.5**	97.5	72.1	**73.0**	**73.0**	72.1	**99.6**	**99.6**	**99.6**	99.2	**84.9**	82.2	81.0	79.6
0.02	**93.7**	90.8	91.3	92.2	91.6	92.6	**97.0**	94.6	60.4	56.5	64.3	**65.6**	94.5	96.5	97.6	**98.0**	**66.7**	64.1	63.9	61.7
0.03	91.3	87.9	88.4	**92.2**	89.7	91.1	92.6	**94.6**	50.4	54.7	**57.3**	52.6	90.3	89.9	**91.0**	89.9	**54.0**	52.6	52.8	51.7
0.05	85.0	85.0	**89.8**	87.9	88.2	92.1	92.1	**94.6**	23.9	28.7	28.7	**29.5**	74.0	79.4	80.6	**84.5**	**37.7**	28.3	26.8	25.9
0.07	81.6	81.6	85.9	**86.9**	78.9	84.8	87.7	**89.7**	10.4	7.3	**20.4**	10.4	**67.4**	58.5	59.6	58.1	–	–	–	–
0.10	84.0	85.5	85.5	**86.9**	73.5	**83.8**	78.4	79.9	3.4	0.4	**19.5**	6.9	37.9	37.6	46.9	**51.9**	–	–	–	–

ϵ	Pendigits				Satimage				ϵ	Compas				German			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$		$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**99.6**	99.2	99.3	99.2	**95.5**	93.8	94.7	**95.5**	0	56.4	62.0	68.0	**70.5**	**85.5**	83.5	82.5	85.0
0.005	**98.4**	98.0	98.3	97.9	84.8	84.7	84.5	**85.2**	0.001	47.9	53.6	58.7	**61.9**	**85.0**	83.5	81.5	84.5
0.01	96.4	96.8	**97.1**	96.8	75.4	77.6	78.9	**81.1**	0.002	44.0	49.5	55.3	**56.8**	**85.0**	82.5	81.5	84.0
0.02	92.7	**93.9**	93.7	93.4	74.9	75.9	79.5	**80.2**	0.005	33.2	38.5	43.7	**47.1**	**84.0**	81.0	81.0	83.5
0.03	88.4	91.5	91.4	**91.9**	66.7	68.3	70.8	**73.3**	0.01	25.1	30.2	31.9	**37.1**	**82.5**	76.5	78.5	82.0
0.05	77.2	83.4	**84.1**	83.7	55.6	57.7	57.5	**60.1**	0.02	17.6	21.5	24.5	**27.1**	**80.0**	76.5	76.0	78.5
0.07	61.8	67.7	**68.4**	**68.4**	49.7	41.0	42.7	**51.5**	0.03	13.5	17.2	19.3	**21.3**	**75.0**	**75.0**	73.0	74.0
0.10	38.4	41.2	49.0	**52.5**	**41.8**	35.0	35.7	35.9	0.05	9.1	12.9	16.1	**16.8**	**70.5**	69.5	63.5	69.0

Table 4 Percentages of provable stability or individual fairness for Intervals $\mathcal{I}$ with Euclidean distance δ_2 on the whole test sets T

ϵ	Australian				BreastCancer				Diabetes				Fourclass				Letter			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	**100**	**100**	**100**	99.5	**100**	99.0	99.5	95.2	96.0	**98.7**	**98.7**	**100**	**100**	**100**	**100**	**97.9**	96.3	95.2	95.6
0.005	97.1	98.5	97.5	**99.0**	96.5	**99.5**	98.0	98.5	78.7	**80.4**	80.0	77.3	**99.6**	**99.6**	99.2	98.8	**91.7**	88.6	85.4	83.0
0.01	95.1	96.6	96.1	**97.1**	96.5	**99.0**	96.5	97.0	**62.6**	59.5	56.5	54.3	99.2	**99.6**	98.4	97.6	**82.6**	75.1	69.4	65.2
0.02	92.7	91.3	91.7	**94.6**	92.1	**93.1**	91.6	**93.1**	**31.3**	25.6	23.9	23.0	**93.4**	91.4	86.8	85.2	**54.6**	45.3	40.2	36.9
0.03	91.3	88.8	89.3	**92.2**	86.7	**88.7**	87.7	88.2	**13.0**	9.1	7.8	6.1	**78.2**	75.5	75.5	72.8	**31.0**	24.8	21.2	19.2
0.05	**85.9**	84.0	85.5	**85.9**	75.0	77.9	81.3	**83.8**	**3.0**	1.7	0.8	0.8	**44.9**	42.6	40.3	37.6	**7.7**	6.6	5.7	5.2
0.07	84.5	82.1	**85.5**	83.5	65.2	65.6	67.6	**74.5**	**1.3**	**1.3**	0	0	**24.0**	23.6	22.8	19.3	–	–	–	–
0.10	82.1	80.1	**85.0**	83.0	37.7	38.7	42.6	**52.9**	**0.8**	**0.8**	0	0	**9.3**	8.5	8.1	8.1	–	–	–	–

ϵ	Pendigits				Satimage			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**99.6**	99.4	99.4	**99.6**	93.8	93.8	93.7	**94.3**
0.005	**98.3**	98.0	98.1	97.8	**73.7**	72.8	72.4	72.3
0.01	**96.6**	96.1	95.7	95.0	**56.0**	54.9	54.0	54.2
0.02	**91.4**	90.5	89.5	89.0	31.4	31.0	31.8	**32.7**
0.03	**82.0**	81.9	81.0	79.9	**19.3**	18.5	18.3	18.6
0.05	58.9	**60.6**	59.7	58.9	9.6	9.6	9.6	**9.8**
0.07	36.3	**38.4**	37.5	36.8	**8.2**	8.1	7.9	7.8
0.10	**13.5**	13.4	13.2	13.2	**4.7**	4.6	4.6	4.5

ϵ	Compas				German			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0	58.2	62.6	69.1	**71.0**	**85.0**	83.0	83.5	82.5
0.001	48.2	54.5	61.1	**62.4**	**85.0**	83.0	83.0	82.0
0.002	45.5	50.9	56.7	**57.6**	**85.0**	82.5	82.5	82.0
0.005	35.2	41.4	45.4	**46.0**	**85.0**	81.5	82.5	81.5
0.01	25.6	30.7	32.8	**36.4**	**83.0**	78.0	81.0	81.5
0.02	17.8	20.8	23.8	**27.4**	**80.0**	75.5	78.0	78.5
0.03	13.5	16.6	19.5	**21.7**	**76.0**	72.5	75.0	73.5
0.05	10.2	12.7	14.3	**16.4**	**72.0**	68.5	69.0	69.5

Table 5 Percentages of provable stability or individual fairness for Zonotopes $\mathcal{I}$ with Euclidean distance δ_2 on the whole test sets T

ϵ	Australian				BreastCancer				Diabetes				Fourclass				Letter			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**100**	**100**	**100**	**100**	99.5	**100**	99.0	99.5	96.5	96.0	**98.7**	**98.7**	**100**	**100**	**100**	**100**	**97.9**	96.3	95.2	95.7
0.005	98.0	**99.0**	98.5	**99.0**	97.5	**99.5**	99.0	99.0	85.2	87.8	**88.2**	87.8	99.6	99.6	**100**	**100**	**94.3**	92.1	91.0	90.1
0.01	97.1	97.1	96.6	**98.0**	96.5	98.5	**99.0**	98.0	79.1	**81.3**	77.3	79.1	99.2	**99.6**	**99.6**	99.2	**88.0**	86.0	84.2	82.8
0.02	95.1	92.2	95.1	**96.6**	94.1	95.1	94.6	**96.5**	66.5	**66.9**	65.2	64.7	90.7	94.5	94.5	**96.5**	**68.3**	66.6	65.0	63.3
0.03	93.7	89.3	93.7	**94.6**	92.6	**97.0**	95.1	94.6	50.0	52.6	55.2	**57.3**	81.4	83.3	84.5	**86.0**	**51.4**	51.3	50.5	49.1
0.05	89.3	87.4	**89.8**	88.8	89.2	93.1	92.6	**95.5**	25.6	30.0	31.7	**38.7**	56.2	64.7	**69.7**	68.9	**28.7**	27.8	27.1	25.8
0.07	85.0	83.0	**87.9**	85.5	77.4	86.7	91.6	**95.1**	10.8	8.2	**11.3**	10.0	36.0	37.2	44.1	**49.2**	–	–	–	–
0.10	82.6	80.1	**85.5**	84.5	66.6	75.9	75.9	**78.9**	4.3	1.7	**14.3**	2.1	19.7	22.0	31.0	**32.1**	–	–	–	–

ϵ	Pendigits				Satimage			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	99.6	99.6	99.4	**99.7**	95.4	95.0	95.2	**96.0**
0.005	**98.8**	98.5	98.6	98.7	84.8	85.1	85.5	**86.8**
0.01	97.6	97.8	**97.9**	97.8	76.1	77.9	78.7	**79.4**
0.02	94.9	95.4	95.4	**95.5**	68.2	70.2	74.4	**76.0**
0.03	91.1	92.7	93.0	**93.1**	60.4	63.0	64.9	**67.8**
0.05	79.7	85.3	**86.7**	85.8	44.4	49.4	52.6	**56.9**
0.07	62.6	71.1	**73.7**	73.5	25.5	31.6	34.8	**42.5**
0.10	38.7	48.3	51.3	**53.1**	9.7	26.6	24.9	**36.2**

ϵ	Compas				German			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0	58.2	62.6	69.1	**71.0**	**85.0**	83.0	83.5	82.5
0.001	49.0	55.5	61.9	**63.4**	**85.0**	83.0	83.0	82.5
0.002	46.8	52.6	58.7	**59.9**	**85.0**	83.0	82.5	82.0
0.005	38.2	45.1	49.6	**52.4**	84.5	83.0	82.5	82.0
0.01	29.8	36.4	40.8	**45.8**	**83.5**	80.0	81.0	81.5
0.02	22.9	28.3	32.3	**35.4**	79.5	77.5	78.5	**80.5**
0.03	17.9	23.1	25.7	**28.6**	76.0	75.0	**77.5**	**77.5**
0.05	12.3	16.7	18.7	**22.0**	72.5	**74.0**	**74.0**	73.0

Table 6 Average certification time per sample in seconds

Dataset	Intervals $\mathcal{I}$		Zonotopes $\mathcal{Z}$	
	δ_1 (s)	δ_2 (s)	δ_1 (s)	δ_2 (s)
Australian	0.01	0.01	0.11	0.22
BreastCancer	0.01	0.01	0.06	0.05
Diabetes	0.11	0.07	0.55	0.58
Fourclass	0.04	0.04	0.35	0.40
Letter	5.44	4.97	21.89	22.64
Pendigits	0.26	0.57	9.99	9.70
Satimage	0.33	2.90	11.91	4.29
Compas	15.30	18.48	140.82	239.99
German	0.75	0.74	5.08	7.67

The fairness experiments show that kNN predictions on:

(i) Compas are rather unfair on the sensitive *race* category, since the average provable race fairness for all k with $\epsilon = 0$ is 64.7%;
(ii) German are rather fair on the sensitive *gender* attribute, since the average provable gender fairness for all k with $\epsilon = 0$ is 83.8%;
(iii) Compas are always more fair with $k = 7$;
(iv) German are mostly more fair with $k = 1$.

Table 6 shows the average certification time, in seconds, per input sample $\mathbf{x}$ and per magnitude ϵ. This is computed as the average time for executing NAVe for all $k \in \{1, 3, 5, 7\}$ on a given input sample (i.e., average on the whole test set T) and for a given magnitude ϵ (i.e., average on the 8 magnitudes ϵ). Our certification technique turns out to be quite fast, where the peak average time of about 4 min is reached for certifying the individual fairness of Compas samples with Euclidean distance through zonotopes, very likely due to one-hot encoding that explodes the number of features from 10 to 370.

6.2.1 Robustness

Table 7 reports the percentages of provable robustness for the interval abstraction $\mathcal{I}$ and Euclidean distance δ_2. Recall from Sect. 2.3 that a classifier is robust when it is both stable and accurate on its input sample, so that the provable robustness inferred by our tool NAVe on a test set T is defined as follows: for all k and ϵ,

$$\text{ProvableRobustness}_{k,\epsilon} \triangleq |\{(\mathbf{x}, l_{\mathbf{x}}) \in T \mid |C^{\mathcal{I}}_{D,\delta_2,k}(P^{\epsilon}_{\infty}(\mathbf{x}))| = 1,\ k\text{NN}(\mathbf{x}) = l_{\mathbf{x}}\}|/|T|.$$

For the sake of comparison with stability, ϵ is limited to 0.05 because for higher thresholds the robustness percentages were too low. Let us recall that provable robustness is necessarily less than or equal to accuracy and provable stability. As expected, robustness behaves similarly to stability, where the relative comparison of Table 7 with stability must consider Table 4. In particular, Australian, Diabetes and Satimage exhibit smaller lower bounds on provable robustness w.r.t. stability, which is due to the lower accuracy of kNN on these datasets (cf. Table 1). This observation turns out to be precise for the dataset Australian, where for $\epsilon = 0.001$ our tool NAVe infers 100% stability for any k (cf. Table 4): Hence, in this case, robustness actually coincides with accuracy, as the lack of accuracy is the sole reason why kNN is not robust. The same effect happens for Diabetes, where, for $\epsilon \leq 0.005$, stability

Table 7 Percentage of provable robustness for Intervals $\mathcal{I}$ with Euclidean distance δ_2 on the whole test sets T

ϵ	Australian				BreastCancer				Diabetes				Fourclass			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	77.8	80.2	**82.6**	**82.6**	92.6	**94.6**	93.6	93.1	69.1	**71.3**	69.6	**71.3**	**100**	**100**	**100**	**100**
0.005	76.8	79.2	81.2	**81.6**	91.2	**94.1**	93.1	92.6	60.0	**62.2**	61.3	59.1	99.6	99.6	99.2	**98.8**
0.01	75.4	77.8	79.7	**80.7**	91.2	**93.6**	91.7	92.2	49.1	**49.6**	47.0	46.5	99.2	**99.6**	98.4	97.7
0.02	73.9	75.8	77.8	**79.2**	88.2	**89.7**	88.7	**89.7**	**25.2**	23.0	22.6	21.7	**93.4**	91.5	86.8	85.3
0.03	72.9	74.4	76.8	**77.8**	84.3	86.8	86.8	**87.3**	**11.7**	8.7	7.4	6.1	**78.3**	75.6	75.6	72.9
0.05	68.6	71.5	**74.4**	73.9	74.5	77.5	80.9	**83.3**	**2.6**	1.7	0.9	0.9	**45.0**	42.6	40.3	37.6

ϵ	Letter				Pendigits				Satimage			
	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$	$k=1$	$k=3$	$k=5$	$k=7$
0.001	**94.7**	93.6	92.9	93.0	**97.6**	97.5	97.4	97.4	86.1	87.4	87.0	**87.5**
0.005	**90.4**	87.8	84.8	82.7	**96.8**	**96.8**	96.7	96.3	**71.7**	**71.7**	71.0	70.8
0.01	**82.2**	75.0	69.4	65.1	**95.6**	95.3	94.9	94.1	**55.4**	54.6	53.8	53.8
0.02	**54.6**	45.4	40.3	37.0	**90.9**	90.2	89.2	88.7	31.4	31.0	31.7	**32.6**
0.03	**31.0**	24.9	21.3	19.2	**81.7**	**81.7**	80.8	79.8	**19.3**	18.6	18.3	18.6
0.05	**7.7**	6.6	5.8	5.2	58.9	**60.6**	59.7	58.9	9.6	9.7	9.7	**9.8**

ranges over 80%, while robustness is around 60%, once again due to lack of accuracy of kNN classification.

7 Conclusion

We have shown how to design an abstract interpretation of k-nearest neighbor classifiers and how this technique defines, to the best of our knowledge, the first robustness certification framework for this popular ML algorithm. We implemented and experimentally evaluated our verification technique. The experiments show that our approach is effective and precise, and that kNN classification is generally robust for numerical perturbations less than $\pm 3\%$.

As any formal verification method, our robustness certification technique is sound, meaning that if a classifier is proved stable over an adversarial region R, then every input in R will actually receive the same classification. However, our certification method, in general, is not complete; namely, the verification may suffer from a precision loss, thus failing to prove stability when this actually holds. This incompleteness makes our verification method susceptible to false negatives, which is the primary limitation of our approach, shared with any incomplete verification method. As discussed in Sect. 3.4, this issue can be mitigated by employing more precise abstract domains to reduce the loss of precision or by partitioning the adversarial region and applying the abstract verification tool to smaller inputs, similarly to the splitting technique described in Sect. 5 for categorical features.

As future work, we plan to design a new numerical abstraction that can precisely track the role of different features when comparing abstract distances between two samples. Ideally, we would aim to achieve a *complete stability certification* of kNN.

Acknowledgements Francesco Ranzato was partially funded by: the *Italian MUR*, under the PRIN 2022 PNRR Project No. P2022HXNSC; *Meta* (formerly *Facebook*) *Research*, under a "Probability and Programming Research Award" and under a *WhatsApp Research Award* on "Privacy-aware Program Analysis"; by an *Amazon Research Award* for "AWS Automated Reasoning." We thank Ahmad Shakeel for his contribution to the data poisoning part in Sect. 4.

Author contributions All the authors equally contributed to this research.

Funding Open access funding provided by Università degli Studi di Padova within the CRUI-CARE Agreement.

Data availibility The source code of our tool together with datasets and scripts for reproducing our experimental results is available on GitHub: https://github.com/abstract-machine-learning/NAVe.

Declarations

Conflict of interest The authors declare no competing interests.

References

1. Albarghouthi A (2021) Introduction to neural network verification. Found Trends Program Lang 7(1–2):1–157. https://doi.org/10.1561/2500000051
2. Altman N (1992) An introduction to kernel and nearest-neighbor nonparametric regression. Am Stat 46(3):175–185. https://doi.org/10.1080/00031305.1992.10475879
3. Amsaleg L, Bailey J, Barbe A et al (2021) High intrinsic dimensionality facilitates adversarial attack: theoretical evidence. IEEE Trans Inf Forensics Secur 16:854–865. https://doi.org/10.1109/TIFS.2020.3023274
4. Bontempi G, Birattari M, Bersini H (1999) Lazy learning for local modelling and control design. Int J Control 72(7–8):643–658. https://doi.org/10.1080/002071799220830
5. Calzavara S, Ferrara P, Lucchese C (2020) Certifying decision trees against evasion attacks by program analysis. In: Proceedings of the 25th European symposium on research in computer security, ESORICS 2020, LNCS, vol 12309. Springer, pp 421–438. https://doi.org/10.1007/978-3-030-59013-0_21
6. Carlini N, Wagner DA (2017) Towards evaluating the robustness of neural networks. In: Proceedings of the IEEE symposium on security and privacy. IEEE S&P, pp 39–57. https://doi.org/10.1109/SP.2017.49
7. Cousot P (2021) Principles of abstract interpretation. MIT Press, Cambridge
8. Cousot P, Cousot R (1977) Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints. In: Proceedings of the 4th ACM symposium on principles of programming languages, POPL 1977, pp 238–252. https://doi.org/10.1145/512950.512973
9. De Figueiredo LH, Stolfi J (2004) Affine arithmetic: concepts and applications. Numer Algorithms 37:147–158. https://doi.org/10.1023/B:NUMA.0000049462.70970.B6
10. Dwork C, Hardt M, Pitassi T et al (2012) Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference, pp 214–226. https://doi.org/10.1145/2090236.2090255
11. Fan AZ, Koutris P (2022) Certifiable robustness for nearest neighbor classifiers. In: Olteanu D, Vortmeier N (eds) Proceedings of the 25th international conference on database theory, ICDT 2022, LIPIcs vol 220, pp 6:1–6:20. https://doi.org/10.4230/LIPIcs.ICDT.2022.6
12. Fassina N, Ranzato F, Zanella M (2023a) NAVe: *k*NN abstract verifier. https://github.com/abstract-machine-learning/NAVe
13. Fassina N, Ranzato F, Zanella M (2023b) Robustness certification of k-nearest neighbors. In: Proceedings of the IEEE international conference on data mining, ICDM 2023. IEEE, pp 110–119. https://doi.org/10.1109/ICDM58522.2023.00020
14. Fishburn PC (1985) Interval orders and interval graphs: a study of partially ordered sets. Wiley, Hoboken
15. Gehr T, Mirman M, Drachsler-Cohen D et al (2018) AI2: safety and robustness certification of neural networks with abstract interpretation. In: Proceedings of the 2018 IEEE symposium on security and privacy. IEEE S&P 2018, pp 3–18. https://doi.org/10.1109/SP.2018.00058
16. Ghorbal K, Goubault E, Putot S (2009) The zonotope abstract domain Taylor1+. In: Proceedings of the 21st international conference on automated verification, CAV 2009. Springer, LNCS vol 5643, pp 627–633. https://doi.org/10.1007/978-3-642-02658-4_47
17. Giacobazzi R, Ranzato F (2022) History of abstract interpretation. IEEE Ann Hist Comput 44(2):33–43. https://doi.org/10.1109/MAHC.2021.3133136
18. Goodfellow I, McDaniel P, Papernot N (2018) Making machine learning robust against adversarial inputs. Commun ACM 61(7):56–66. https://doi.org/10.1145/3134599
19. Goubault E, Putot S (2015) A zonotopic framework for functional abstractions. Formal Methods Syst Des 47(3):302–360. https://doi.org/10.1007/s10703-015-0238-z
20. Jia J, Liu Y, Cao X et al (2022) Certified robustness of nearest neighbors against data poisoning and backdoor attacks. In: Proceedings of the 36th AAAI conference on artificial intelligence, pp 9575–9583. https://ojs.aaai.org/index.php/AAAI/article/view/21191
21. Kramer O (2013) K-nearest neighbors. In: Dimensionality reduction with unsupervised nearest neighbors, intelligent systems reference library, vol 51. Springer, Berlin, pp 13–23. https://doi.org/10.1007/978-3-642-38652-7_2
22. Kurakin A, Goodfellow IJ, Bengio S (2017) Adversarial machine learning at scale. In: Proceedings of the 5th international conference on learning representations, ICLR 2017. https://openreview.net/forum?id=BJm4T4Kgx
23. Li Y, Wang J, Wang C (2022) Proving robustness of KNN against adversarial data poisoning. In: Proceedings of the 22nd international conference on formal methods in computer-aided design, FMCAD 2022. IEEE, pp 7–16. https://doi.org/10.34727/2022/isbn.978-3-85448-053-2_6
24. Li Y, Wang J, Wang C (2023a) Certifying the fairness of KNN in the presence of dataset bias. In: Proceedings of the 35th international conference on computer aided verification, CAV 2023, lecture notes in computer science, vol 13965. Springer, pp 335–357. https://doi.org/10.1007/978-3-031-37703-7_16

25. Li Y, Wang J, Wang C (2023b) Systematic testing of the data-poisoning robustness of KNN. In: Proceedings of the 32nd ACM SIGSOFT international symposium on software testing and analysis, ISSTA 2023. ACM, pp 1207–1218. https://doi.org/10.1145/3597926.3598129
26. Liu C, Arnon T, Lazarus C et al (2021) Algorithms for verifying deep neural networks. Found Trends Optim 4(3–4):244–404. https://doi.org/10.1561/2400000035
27. Liu Y, Peng J, Chen L et al (2020) Abstract interpretation based robustness certification for graph convolutional networks. In: Proceedings of the 24th European conference on artificial intelligence, ECAI 2020, pp 1309–1315. https://doi.org/10.3233/FAIA200233
28. McDaniel PD, Papernot N, Celik ZB (2016) Machine learning in adversarial settings. IEEE Secur Priv 14(3):68–72. https://doi.org/10.1109/MSP.2016.51
29. Mehrabi N, Morstatter F, Saxena N et al (2022) A survey on bias and fairness in machine learning. ACM Comput Surv 54(6):115:1-115:35. https://doi.org/10.1145/3457607
30. Miné A (2006) The octagon abstract domain. High Order Symb Comput 19(1):31–100. https://doi.org/10.1007/S10990-006-8609-1
31. Miné A (2017) Tutorial on static inference of numeric invariants by abstract interpretation. Found Trends Program Lang 4(3–4):120–372. https://doi.org/10.1561/2500000034
32. Mirman M, Gehr T, Vechev MT (2018) Differentiable abstract interpretation for provably robust neural networks. In: Proceedings of the 35th international conference on machine learning, ICML 2018, pp 3575–3583. http://proceedings.mlr.press/v80/mirman18b.html
33. Pal A, Ranzato F, Urban C et al (2024) Abstract interpretation-based feature importance for support vector machines. In: Proceedings of the international conferenc on verification, model checking, and abstract interpretation, VMCAI 2024. Springer LNCS 14499, pp 27–49. https://doi.org/10.1007/978-3-031-50524-9_2
34. Ranzato F, Zanella M (2019) Robustness verification of support vector machines. In: Proceedings of the 26th international static analysis symposium, SAS 2019, LNCS vol 11822, pp 271–295. https://doi.org/10.1007/978-3-030-32304-2_14
35. Ranzato F, Zanella M (2020) Abstract interpretation of decision tree ensemble classifiers. In: Proceedings of the thirty-fourth aaai conference on artificial intelligence, AAAI 2020, pp 5478–5486. https://aaai.org/ojs/index.php/AAAI/article/view/5998
36. Ranzato F, Zanella M (2021) Genetic adversarial training of decision trees. In: Proceedings of the 2021 genetic and evolutionary computation conference, GECCO 2021. ACM, pp 358–367. https://doi.org/10.1145/3449639.3459286
37. Ranzato F, Urban C, Zanella M (2021) Fairness-aware training of decision trees by abstract interpretation. In: Proceedings of the 30th ACM international conference on information and knowledge management, CIKM, pp 1508–1517. https://doi.org/10.1145/3459637.3482342
38. Ruoss A, Balunovic M, Fischer M et al (2020) Learning certified individually fair representations. In: Proceedings of the 34th annual conference on advances in neural information processing systems, NeurIPS 2020. https://proceedings.neurips.cc/paper/2020/hash/55d491cf951b1b920900684d71419282-Abstract.html
39. Singh G, Gehr T, Püschel M et al (2019) An abstract domain for certifying neural networks. Proc ACM Program Lang 3:41:1-41:30. https://doi.org/10.1145/3290354
40. Singh G, Gehr T, Püschel M et al (2019b) Boosting robustness certification of neural networks. In: Proceedings of the of the 7th international conference on learning representations, ICLR. https://openreview.net/forum?id=HJgeEh09KQ
41. Sitawarin C, Wagner DA (2020) Minimum-norm adversarial examples on KNN and KNN based models. In: Proceedings of the IEEE security and privacy workshops, SP workshops, 2020. IEEE, pp 34–40. https://doi.org/10.1109/SPW50608.2020.00023
42. Sitawarin C, Kornaropoulos EM, Song D et al (2021) Adversarial examples for k-nearest neighbor classifiers based on higher-order Voronoi diagrams. In: Proceedings of the annual conference on neural information processing systems, NeurIPS, pp 15486–15497. https://proceedings.neurips.cc/paper/2021/hash/82ca5dd156cc926b2992f73c2896f761-Abstract.html
43. Tian Z, Cui L, Liang J et al (2023) A comprehensive survey on poisoning attacks and countermeasures in machine learning. ACM Comput Surv 55(8):166:1-166:35. https://doi.org/10.1145/3551636
44. Urban C, Miné A (2021) A review of formal methods applied to machine learning. CoRR arXiv:2104.02466
45. Wang L, Liu X, Yi J et al (2019) Evaluating the robustness of nearest neighbor classifiers: a primal-dual perspective. arXiv:1906.03972
46. Wang Y, Jha S, Chaudhuri K (2018) Analyzing the robustness of nearest neighbors to adversarial examples. In: Proceedings of the 35th international conference on machine learning, ICML, pp 5120–5129. http://proceedings.mlr.press/v80/wang18c.html

47. Wang Z, Ma J, Wang X et al (2023) Threats to training: a survey of poisoning attacks and defenses on machine learning systems. ACM Comput Surv 55(7):134:1-134:36. https://doi.org/10.1145/3538707
48. Yang Y, Rashtchian C, Wang Y et al (2020) Robustness for non-parametric classification: a generic attack and defense. In: Proceedings of the 23rd international conferenc on artificial intelligence and statistics, AISTATS, PMLR vol 108, pp 941–951. http://proceedings.mlr.press/v108/yang20b.html

Nicolò Fassina is a Microsoft Dynamics 365 Developer who earned his MSc degree in Computer Science from the Department of Mathematics "Tullio Levi-Civita" at the University of Padova, Padua, Italy. Throughout his academic studies, he delved deep into abstract interpretation techniques, particularly focusing on their application in formally verifying machine learning methods.

Francesco Ranzato is a Professor of Computer Science with the Department of Mathematics "Tullio Levi-Civita," University of Padova, Padua, Italy. Since 1994, he has been working on abstract interpretation principles and applications. He is the corresponding author of this article.

Marco Zanella earned a PhD degree in Computer Science at the Department of Mathematics "Tullio Levi-Civita," University of Padova, Padua, Italy. Since 2018, he has been working on the applications of abstract interpretation to machine learning models.