

UNIVERSITY OF CALIFORNIA SAN DIEGO

The Primacy of Applied Privacy

A dissertation submitted in partial satisfaction of the
requirements for the degree Doctor of Philosophy

in

Computer Science

by

Casey Meehan

Committee in charge:

Professor Kamalika Chaudhuri, Chair
Professor Taylor Berg-Kirkpatrick
Professor Sanjoy Dasgupta
Professor Alon Orlitsky

2023

Copyright
Casey Meehan, 2023
All rights reserved.

The Dissertation of Casey Meehan is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2023

DEDICATION

The fact that I have made something I can write a dedication for is owed all to my parents. I cannot imagine following my heart these past few years without their unrelenting support and encouragement.

TABLE OF CONTENTS

Dissertation Approval Page	iii
Dedication	iv
Table of Contents	v
List of Figures	vii
List of Tables	viii
Acknowledgements	ix
Vita	x
Abstract of the Dissertation	xi
Chapter 1 When are Non-Parametric Methods Robust?	1
1.1 Introduction	1
1.1.1 Related Work	3
1.2 Preliminaries	4
1.2.1 Setting	4
1.2.2 Notions of Consistency	5
1.2.3 Non-parametric Classifiers	8
1.3 Warm Up: r -separated distributions	10
1.4 General Distributions	14
1.4.1 The r -Optimal Classifier and Adversarial Pruning	14
1.4.2 Convergence Guarantees	15
1.5 Validation	17
1.5.1 Experimental Setup	18
1.5.2 Results	19
1.5.3 Discussion	19
1.6 Conclusion	20
Chapter 2 Consistent Non-Parametric Methods for Maximizing Robustness	21
2.1 Introduction	21
2.2 Preliminaries	24
2.3 The Neighborhood preserving Bayes optimal classifier	26
2.3.1 Neighborhood Consistency	29
2.4 Neighborhood Consistent Non-Parametric Classifiers	30
2.4.1 Splitting Numbers	31
2.4.2 Sufficient Conditions for Neighborhood Consistency	32
2.4.3 Nearest Neighbors and Kernel Classifiers	33
2.4.4 Histogram Classifiers	34

2.5	Validation	35
2.6	Related Work	37
Chapter 3	Sample Complexity of Robust Linear Classification on Separated Data	39
3.1	Introduction	39
3.1.1	Related Work	41
3.2	Preliminaries	43
3.2.1	Standard and Robust Loss	44
3.2.2	Expected Loss and Sample Complexity	44
Bibliography	46

LIST OF FIGURES

Figure 1.1.	H_S is astute in the green region, but not robust in the red region.	10
Figure 1.2.	Empirical accuracy/astuteness of different classifiers as a function of training sample size. Accuracy is shown in green, astuteness in purple. Left : Noiseless Setting. Right: Noisy Setting. Top Row: Histogram Classifier, Bottom Row: 1-Nearest Neighbor	17
Figure 2.1.	A data distribution demonstrating the difficulties with fixed radius balls for robustness regions. The red represents negatively labeled points, and the blue positive. If the robustness radius is set too large (panel (a)), then the regions of A and B intersect leading to a loss of accuracy. If the radius is set too small (panel (b)), this leads to a loss of robustness at point C where in principle it should be possible to defend against a larger amount of adversarial attacks.	22
Figure 2.2.	The decision boundary of the neighborhood preserving Bayes optimal classifier is shown in green, and the neighborhood preserving robust region of x is shown in pink. The former consists of points equidistant from μ^+, μ^- , and the latter consists of points equidistant from x, μ^+	27
Figure 2.3.	we have a histogram classifier being applied to the blue and red regions. The classifier will be unable to construct good labels in the cells labeled A, B, C , and consequently will not be robust with respect to V_x^κ for sufficiently large κ	34
Figure 2.4.	Plots of astuteness against the training sample size. In both panels, accuracy is plotted in red, and the varying levels of robustness regions ($\kappa = 0.1, 0.3, 0.5$) are given in blue, green and purple. In panel (a), observe that as sample size increases, every measure of astuteness converges towards 0.8 which is as predicted by Corollary 33. In panel (b), although the accuracy appears to converge, none of the robustness measure. In fact, they get progressively worse the larger κ gets.	35

LIST OF TABLES

ACKNOWLEDGEMENTS

VITA

2015	Bachelor of Science, Brown University
2018	Master of Science, Harvard University
2023	Doctor of Philosophy, University of California, San Diego

ABSTRACT OF THE DISSERTATION

The Primacy of Applied Privacy

by

Casey Meehan

Doctor of Philosophy in Computer Science

University of California San Diego, 2023

Professor Kamalika Chaudhuri, Chair

As data collection for machine learning (ML) tasks has become more pervasive, it has also become more heterogeneous: we share our writing, images, voices, and location online every day. Naturally, the associated privacy risks are just as complex and variable. My research advances practical data privacy through two avenues: 1) drafting provable privacy definitions and mechanisms for safely sharing data in different ML domains, and 2) empirically quantifying how ML models memorize their sensitive training data and thereby risk disclosing it. This dissertation details the various data domains/tasks considered, and the corresponding privacy methods proposed.

Chapter 1

When are Non-Parametric Methods Robust?

1.1 Introduction

Recent work has shown that many classifiers tend to be highly non-robust and that small strategic modifications to regular test inputs can cause them to misclassify [1, 2, 3]. Motivated by the use of machine learning in safety-critical applications, this phenomenon has recently received considerable interest; however, what exactly causes this phenomenon – known in the literature as *adversarial examples* – still remains a mystery.

Prior work has looked at three plausible reasons why adversarial examples might exist. The first, of course, is the possibility that in real data distributions, different classes are very close together in space – which does not seem plausible in practice. Another possibility is that classification algorithms may require more data to be robust than to be merely accurate; some prior work [4, 5, 6] suggests that this might be true for certain classifiers or algorithms. Finally, others [7, 8, 5] have suggested that better training algorithms may give rise to more robust classifiers – and that in some cases, finding robust classifiers may even be computationally challenging.

In this work, we consider this problem in the context of general non-parametric classifiers. Contrary to parametrics, non-parametric methods are a form of local classifiers, and include a large number of pattern recognition methods such as nearest neighbors, decision trees, random

forests and kernel classifiers. There is a richly developed statistical theory of non-parametric methods [9], which focuses on accuracy, and provides very general conditions under which these methods converge to the Bayes optimal with growing number of samples. We, in contrast, analyze robustness properties of these methods, and ask instead when they converge to the classifier with the highest astuteness at a desired radius r . Recall that the astuteness of a classifier at radius r is the fraction of points from the distribution on which it is accurate and has the same prediction up to a distance r [5, 4].

We begin by looking at the very simple case when data from different classes is well-separated – by at least a distance $2r$. Although achieving astuteness in this case may appear trivial, we show that even in this highly favorable case, not all non-parametric methods provide robust classifiers – and this even holds for methods that converge to the Bayes optimal in the large sample limit.

This raises the natural question – when do non-parametric methods produce astute classifiers? We next provide conditions under which a non-parametric method converges to the most astute classifier in the large sample limit under well-separated data. Our conditions are analogous to the classical conditions for convergence to the Bayes optimal [9, 10], but a little stronger. We show that nearest neighbors and kernel classifiers whose kernel functions decay fast enough, satisfy these conditions, and hence converge to astute classifiers in the large sample limit. In contrast, histogram classifiers, which do converge to the Bayes optimal in the large sample limit, may not converge to the most astute classifier. This indicates that there may be some non-parametric methods, such as nearest neighbors and kernel classifiers, that are more naturally robust when trained on well-separated data, and some that are not.

What happens when different classes in the data are not as well-separated? For this case, [11] proposes a method called Adversarial Pruning that preprocesses the training data by retaining the maximal set of points such that different classes are distance $\geq 2r$ apart, and then trains a non-parametric method on the pruned data. We next prove that if a non-parametric method has certain properties, then the classifier produced by Adversarial Pruning followed by

the method does converge to the most astute classifier in the large sample limit. We show that again nearest neighbors and kernel classifiers whose kernel functions decay faster than inverse polynomials satisfy these properties. Our results thus complement and build upon the empirical results of [11] by providing a performance guarantee.

What can we conclude about the cause for adversarial examples? Our results seem to indicate that at least for non-parametrics, it is mostly the training algorithms that are responsible. With a few exceptions, decades of prior work in machine learning and pattern recognition has largely focussed on designing training methods that provide increasingly accurate classifiers – perhaps to the detriment of other aspects such as robustness. In this context, our results serve to (a) provide a set of guidelines that can be used for designing non-parametric methods that are robust and accurate on well-separated data and (b) demonstrate that when data is not well-separated, preprocessing through adversarial pruning [11] may be used to ensure convergence to optimally astute solutions in the large sample limit.

1.1.1 Related Work

There is a large body of work on adversarial attacks [12, 13, 14, 15, 1] and defenses [16, 17, 18, 19, 20, 21] in the parametric setting, specifically focusing on neural networks. On the other hand, adversarial examples for nonparametric classifiers have mostly been studied in a much more ad-hoc manner, and to our knowledge, there has been no theoretical investigation into general properties of algorithms that promote robustness in non-parametric classifiers.

For nearest neighbors, there has been some prior work on adversarial attacks [22, 23, 5, 11] as well as defenses. Wang et. al. [5] proposes a defense for 1-NN by pruning the input sample. However, their defense learns a classifier whose robustness regions converge towards those of the Bayes optimal classifier, which itself may potentially have poor robustness properties. Yang et. al. [11] accounts for this problem by proposing the notion of the r -optimal classifier, and propose an algorithm called Adversarial Pruning which can be interpreted as a finite sample approximation to the r -optimal. However, they do not provide formal performance guarantees

for Adversarial Pruning, which we do.

For Kernel methods, Hein and Andriushchenko [16] study lower bounds on the norm of the adversarial manipulation that is required for changing a classifiers output. They specifically study bounds for Kernel Classifiers, and propose an empirically based regularization idea that improves robustness. In this work, we improve the robustness properties of kernel classification through adversarial pruning, and show formal guarantees regarding convergence towards the r -optimal classifier.

For decision trees and random forests, attacks and defenses have been provided by [24, 25, 26]. Again, most of the work here is empirical in nature, and convergence guarantees are not provided.

Pruning has a long history of being applied for improving nearest neighbors [27, 28, 29, 30, 31, 32], but this has been entirely done in the context of generalization, without accounting for robustness. In their work, Yang et. al. empirically show that adversarial pruning can improve robustness for nearest neighbor classifiers. However, they do not provide any formal guarantees for their algorithms. In this work, we prove formal guarantees for *adversarial pruning* in the large sample limit, both for nearest neighbors as well as for more general *weight functions*.

There is a long history of literature for understanding the consistency of Kernel classifiers [33, 10], but this has only been done for accuracy and generalization. In this work, we find different conditions are needed to ensure that a Kernel classifier converges in robustness in addition to accuracy.

1.2 Preliminaries

1.2.1 Setting

We consider binary classification where instances are drawn from a totally bounded metric space \mathcal{X} that is equipped with distance metric denoted by d , and the label space is $\{\pm 1\} = \{-1, +1\}$. The classical goal of classification is to build a highly *accurate* classifier,

which we define as follows.

Definition 1. (Accuracy) Let \mathcal{D} be a distribution over $\mathcal{X} \times \{\pm 1\}$, and let $f \in \{\pm 1\}^{\mathcal{X}}$ be a classifier. Then the **accuracy** of f over \mathcal{D} , denoted $A(f, \mathcal{D})$, is the fraction of examples $(x, y) \sim \mathcal{D}$ for which $f(x) = y$. Thus

$$A(f, \mathcal{D}) = P_{(x,y) \sim \mathcal{D}}[f(x) = y].$$

In this work, we consider *robustness* in addition to accuracy. Let $B(x, r)$ denoted the closed ball of radius r centered at x .

Definition 2. (Robustness) A classifier $f \in \{\pm 1\}^{\mathcal{X}}$ is said to be **robust** at x with radius r if $f(x) = f(x')$ for all $x' \in B(x, r)$.

Our goal is to find non-parametric algorithms that output classifiers that are robust, in addition to being accurate. To account for both criteria, we combine them into a notion of *astuteness* [5, 4].

Definition 3. (Astuteness) A classifier $f \in \{\pm 1\}^{\mathcal{X}}$ is said to be **astute** at (x, y) with radius r if f is robust at x with radius r and $f(x) = y$. The **astuteness** of f over \mathcal{D} , denoted $A_r(f, \mathcal{D})$, is the fraction of examples $(x, y) \sim \mathcal{D}$ for which f is astute at (x, y) with radius r . Thus

$$A_r(f, \mathcal{D}) = P_{(x,y) \sim \mathcal{D}}[f(x') = y, \forall x' \in B(x, r)].$$

It is worth noting that $A_0(f, \mathcal{D}) = A(f, \mathcal{D})$, since astuteness with radius 0 is simply the accuracy. For this reason, we will use $A_0(f, \mathcal{D})$ to denote accuracy from this point forwards.

1.2.2 Notions of Consistency

Traditionally, a classification algorithm is said to be consistent if as the sample size grows to infinity, the accuracy of the classifier it learns converges towards the best possible accuracy on the underlying data distribution. We next introduce and formalize an alternative form of consistency, called *r-consistency*, that applies to robust classifiers.

We begin with a formal definition of the Bayes Optimal Classifier – the most accurate classifier on a distribution – and consistency.

Definition 4. (*Bayes Optimal Classifier*) The **Bayes Optimal Classifier** on a distribution \mathcal{D} , denoted by g^* , is defined as follows. Let $\eta(x) = p_{\mathcal{D}}(+1|x)$. Then

$$g^*(x) = \begin{cases} +1 & \eta(x) \geq 0.5 \\ -1 & \eta(x) < 0.5 \end{cases}$$

It can be shown that g^* achieves the highest accuracy over \mathcal{D} over all classifiers.

Definition 5. (*Consistency*) Let M be a classification algorithm over $\mathcal{X} \times \{\pm 1\}$. M is said to be **consistent** if for any \mathcal{D} over $\mathcal{X} \times \{\pm 1\}$, and any ϵ, δ over $(0, 1)$, there exists N such that for $n \geq N$, with probability $1 - \delta$ over $S \sim \mathcal{D}^n$, we have:

$$A(M(S), \mathcal{D}) \geq A(g^*, \mathcal{D}) - \epsilon,$$

where g^* is the Bayes optimal classifier for \mathcal{D} .

How can we incorporate robustness in addition to accuracy in this notion? A plausible way, as used in [5], is that the classifier should converge towards being astute where the Bayes Optimal classifier is astute. However, the Bayes Optimal classifier is not necessarily the most astute classifier and may even have poor astuteness. To see this, consider the following example.

Example 1

Consider \mathcal{D} over $\mathcal{X} = [0, 1]$ such that $\mathcal{D}_{\mathcal{X}}$ is the uniform distribution and

$$p(y = 1|x) = \frac{1}{2} + \sin \frac{4\pi x}{r}.$$

For any point x , there exists $x_1, x_2 \in ([x - r, x + r] \cap [0, 1])$ such that $p(y = 1|x_1) > \frac{1}{2}$ and $p(y = 1|x_2) < \frac{1}{2}$. $A_r(g^*, r) = 0$. However, the classifier that always predicts $f(x) = +1$ does

better. It is robust everywhere, and since $P_{(x,y) \sim \mathcal{D}}[y = +1] = \frac{1}{2}$, it follows that $A_r(f, \mathcal{D}) = \frac{1}{2}$.

This motivates the notion of the r -optimal classifier, introduced by [11], which is the classifier with maximum astuteness.

Definition 6. (*r-optimal classifier*) The ***r-optimal classifier*** of a distribution G denoted by g_r^* is the classifier with maximum astuteness. Thus

$$g_r^* = \arg \max_{f \in \{\pm 1\}^{\mathcal{X}}} A_r(f, \mathcal{D}).$$

We let $A_r^*(\mathcal{D})$ denote $A_r(g_r^*, \mathcal{D})$.

Observe that g_r^* is not necessarily unique. To account for this, we use $A_r^*(\mathcal{D})$ in our definition for r -consistency.

Definition 7. (*r-consistent*) Let M be a classification algorithm over $\mathcal{X} \times \{\pm 1\}$. M is said to be ***r-consistent*** if for any \mathcal{D} , any $\varepsilon, \delta \in (0, 1)$, and $0 < \gamma < r$, there exists N such that for $n \geq N$, with probability $1 - \delta$ over $S \sim \mathcal{D}^n$,

$$A_{r-\gamma}(M(S), \mathcal{D}) \geq A_r^*(\mathcal{D}) - \varepsilon.$$

if the above conditions hold for a specific distribution \mathcal{D} , we say that M is *r-consistent with respect to \mathcal{D}* .

Observe that in addition to the usual ε and δ , there is an extra parameter γ which measures the gap in the robustness radius. We may need this parameter as when classes are exactly $2r$ apart, we may not be able to find the exact robust boundary with only finite samples.

Our analysis will be centered around understanding what kinds of algorithms M provide highly astute classifiers for a given radius r . We begin by first considering the special case of

r -separated distributions.

Definition 8. (r -separated distributions) A distribution \mathcal{D} is said to be **r -separated** if there exist subsets $T^+, T^- \subset \mathcal{X}$ such that

1. $\mathbb{P}_{(x,y) \sim \mathcal{D}}[x \in T^+] = 1$.
2. $\forall x_1 \in T^+, \forall x_2 \in T^-, d(x_1, x_2) > 2r$.

Observe that if \mathcal{D} is r -separated, $A_r(g_r^*, \mathcal{D}) = 1$.

1.2.3 Non-parametric Classifiers

Many non-parametric algorithms classify points by averaging labels over a local neighborhood from their training data. A very general form of this idea is encapsulated in *weight functions* – which is the general form we will use.

Definition 9. [9] A **weight function** W is a non-parametric classifier with the following properties.

1. Given input $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathcal{D}^n$, W constructs functions $w_1^S, \dots, w_n^S : \mathcal{X} \rightarrow [0, 1]$ such that for all $x \in \mathcal{X}$, $\sum_1^n w_i^S(x) = 1$. The functions w_i^S are allowed to depend on x_1, x_2, \dots, x_n but must be independent of y_1, y_2, \dots, y_n .
2. W has output W_S defined as

$$W_S(x) = \begin{cases} +1 & \sum_1^n w_i^S(x) y_i > 0 \\ -1 & \sum_1^n w_i^S(x) y_i \leq 0 \end{cases}$$

As a result, $w_i^S(x)$ can be thought of as the weight that (x_i, y_i) has in classifying x .

Weight functions encompass a fairly extensive set of common non-parametric classifiers, which is the motivation for considering them. We now define several common non-parametric algorithms that can be construed as weight functions.

Definition 10. A *histogram classifier*, H , is a non-parametric classification algorithm over $\mathbb{R}^d \times \{\pm 1\}$ that works as follows. For a distribution \mathcal{D} over $\mathbb{R} \times \{\pm 1\}$, H takes $S = \{(x_i, y_i) : 1 \leq i \leq n\} \sim \mathcal{D}^n$ as input. Let k_i be a sequence with $\lim_{i \rightarrow \infty} k_i = \infty$ and $\lim_{i \rightarrow \infty} \frac{k_i}{i} = 0$. H constructs a set of hypercubes $C = \{c_1, c_2, \dots, c_m\}$ as follows:

1. Initially $C = \{c\}$, where $S \subset c$.
2. For $c \in C$, if c contains more than k_n points of S , then partition c into 2^d equally sized hypercubes, and insert them into C .
3. Repeat step 2 until all cubes in C have at most k_n points.

For $x \in \mathbb{R}$ let $c(x)$ denote the unique cell in C containing x . If $c(x)$ doesn't exist, then $H_S(x) = -1$ by default. Otherwise,

$$H_S(x) = \begin{cases} +1 & \sum_{x_i \in c(x)} y_i > 0 \\ -1 & \sum_{x_i \in c(x)} y_i \leq 0 \end{cases}.$$

Histogram classifiers are weight functions in which all x_i contained within the same cell as x are given the same weight $w_i^S(x)$ in predicting x , while all other x_i are given weight 0.

Definition 11. A *kernel classifier* is a weight function W over $\mathcal{X} \times \{\pm 1\}$ constructed from function $K : \mathbb{R}^+ \cup \{0\} \rightarrow \mathbb{R}^+$ and some sequence $\{h_n\} \subset \mathbb{R}^+$ in the following manner. Given $S = \{(x_i, y_i)\} \sim \mathcal{D}^n$, we have

$$w_i^S(x) = \frac{K(\frac{d(x, x_i)}{h_n})}{\sum_{j=1}^n K(\frac{d(x, x_j)}{h_n})}.$$

Then, as above, W has output

$$W_S(x) = \begin{cases} +1 & \sum_1^n w_i^S(x) y_i > 0 \\ -1 & \sum_1^n w_i^S(x) y_i \leq 0 \end{cases}$$

Finally, we note that k_n -nearest neighbors is also a weight function; $w_i^S(x) = \frac{1}{k_n}$ if x_i is one of the k_n closest neighbors of x and 0 otherwise.

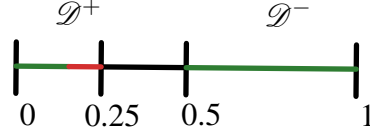


Figure 1.1. H_S is astute in the green region, but not robust in the red region.

1.3 Warm Up: r -separated distributions

We begin by considering the case when the data distribution is r -separated; the more general case is considered in Section 1.4. While classifying r -separated distributions robustly may appear almost trivial, learning an arbitrary classifier does not necessarily produce an astute result. To see this, consider the following example of a histogram classifier – which is known to be consistent.

We let H denote the histogram classifier over \mathbb{R} .

Example 2

Consider the data distribution $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-$ where \mathcal{D}^+ is the uniform distribution over $[0, \frac{1}{4})$ and \mathcal{D}^- is the uniform distribution over $(\frac{1}{2}, 1]$, $p(+1|x) = 1$ for $x \in \mathcal{D}^+$, and $p(-1|x) = 1$ for $x \in \mathcal{D}^-$.

We make the following observations (refer to Figure 1.1).

1. \mathcal{D} is 0.1-separated, since the supports of \mathcal{D}^+ and \mathcal{D}^- have distance $0.25 > 0.2$.
2. If n is sufficiently large, H will construct the cell $[0.25, 0.5)$, which will not be split because it will never contain any points.
3. $H_S(x) = -1$ for $x \in [0.25, 0.5)$.
4. H_S is not astute at $(x, 1)$ for $x \in (0.15, 0.25)$. Thus $A_{0.1}(H_S, \mathcal{D}) = 0.8$.

Example 2 shows that histogram classifiers do not always learn astute classifiers even

when run on r -separated distributions. This motivates the question: which non-parametric classifiers do?

We answer this question in the following theorem, which gives sufficient conditions for a weight function (definition 9) to be r -consistent over an r -separated distribution.

Theorem 12. *Let \mathcal{D} be a distribution over $\mathcal{X} \times \{\pm 1\}$, and let W be a weight function. Let X be a random variable with distribution $\mathcal{D}_{\mathcal{X}}$, and $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathcal{D}^n$. Suppose that for any $0 < a < b$,*

$$\lim_{n \rightarrow \infty} \mathbb{E}_{X, S} \left[\sup_{x' \in B(X, a)} \sum_{i=1}^n w_i^S(x') I_{||x_i - x'|| > b} \right] = 0.$$

Then if \mathcal{D} is r -separated, W is r -consistent with respect to \mathcal{D} .

First, we compare Theorem 12 to Stone's theorem [10], which gives sufficient conditions for a weight function to be consistent (i.e. converge in accuracy towards the Bayes optimal). For convenience, we include a statement of Stone's theorem.

Theorem 13. [10] *Let W be weight function over $\mathcal{X} \times \{\pm 1\}$. Suppose the following conditions hold for any distribution \mathcal{D} over $\mathcal{X} \times \{\pm 1\}$. Let X be a random variable with distribution $\mathcal{D}_{\mathcal{X}}$, and $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathcal{D}^n$. All expectations are taken over X and S .*

1. *There is a constant c such that, for every nonnegative measurable function f satisfying*

$$\mathbb{E}[f(X)] < \infty,$$

$$\mathbb{E} \left[\sum_{i=1}^n w_i^S(X) f(x_i) \right] \leq c \mathbb{E}[f(x)].$$

2. *For all $a > 0$,*

$$\lim_{n \rightarrow \infty} \mathbb{E} \left[\sum_{i=1}^n w_i^S(x) I_{||x_i - X|| > a} \right] = 0,$$

where $I_{||x_i - X|| > a}$ is an indicator variable.

- 3.

$$\lim_{n \rightarrow \infty} \mathbb{E} \left[\max_{1 \leq i \leq n} w_i^S(X) \right] = 0.$$

Then W is consistent.

There are two main differences between Theorem 12 and Stone's theorem.

1. Conditions 1. and 3. of Stone's theorem are no longer necessary. This is because r -separated distributions are well-separated and thus have simpler conditions for consistency. In fact, a slight modification of the arguments of [10] shows that for r -separated distributions, condition 2. alone is sufficient for consistency.
2. Condition 2. is strengthened. Instead of requiring the weight of x_i 's outside of a given radius to go to 0 for $X \sim \mathcal{D}$, we require the same to *uniformly* hold over a ball centered at X .

Theorem 12 provides a general condition that allows us to verify the r -consistency of non-parametric methods. We now show below that two common non-parametric algorithms – k_n -nearest neighbors and kernel classifiers with rapidly decaying kernel functions – satisfy the conditions of Theorem 12.

Corollary 14. *Let \mathcal{D} be any r -separated distribution. Let k_n be any sequence such that $\lim_{n \rightarrow \infty} \frac{k_n}{n} = 0$, and let M be the k_n -nearest neighbors classifier on a sample $S \sim \mathcal{D}^n$. Then M is r -consistent with respect to \mathcal{D} .*

Remarks:

1. Because the data distribution is r -separated, $k_n = 1$ will be r -consistent. Also observe that for r -separated distributions, $k_n = 1$ will converge towards the Bayes Optimal classifier.
2. In general, M converges towards the Bayes Optimal classifier provided that $k_n \rightarrow \infty$ in addition to $k_n/n \rightarrow 0$. This condition is not necessary for r -consistency– because the distribution is r -separated.

We next show that kernel classifiers are also r -consistent on r -separated data distributions, provided the kernel function decreases rapidly enough.

Corollary 15. *Let W be a kernel classifier over $\mathcal{X} \times \{\pm 1\}$ constructed from K and h_n . Suppose the following properties hold for K and h_n .*

1. *For any $c > 1$, $\lim_{x \rightarrow \infty} \frac{K(cx)}{K(x)} = 0$.*
2. *$\lim_{n \rightarrow \infty} h_n = 0$.*

If \mathcal{D} is an r -separated distribution over $\mathcal{X} \times \{\pm 1\}$, then W is r -consistent with respect to \mathcal{D} .

Observe that Condition 1. is satisfied for any $K(x)$ that decreases more rapidly than an inverse polynomial – and is hence satisfied by most popular kernels like the Gaussian kernel. Is the condition on K in Corollary 15 necessary? The following example illustrates that a kernel classifier with any arbitrary K is not necessarily r -consistent. This indicates that some sort of condition needs to be imposed on K to ensure r -consistency; finding a tight necessary condition however is left for future work.

Example 3

Let $\mathcal{X} = [-1, 1]$ and let \mathcal{D} be a distribution with $p_{\mathcal{D}}(-1, -1) = 0.1$ and $p_{\mathcal{D}}(1, 1) = 0.9$. Clearly, \mathcal{D} is 0.3-separated. Let $K(x) = e^{-\min(|x|, 0.2)^2}$. Let h_n be any sequence with $\lim_{n \rightarrow \infty} h_n = 0$ and $\lim_{n \rightarrow \infty} nh_n = \infty$. Let W be the weight classifier with input $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ such that

$$w_i^S(x) = \frac{K\left(\frac{|x - x_i|}{h_n}\right)}{\sum_{j=1}^n K\left(\frac{|x - x_j|}{h_n}\right)}.$$

W can be shown to satisfy all the conditions of Theorem 13 (the proof is analogous to the case for a Gaussian Classifier), and is therefore consistent. However, W does not learn a robust classifier on \mathcal{D} for $r = 0.3$.

Consider $x = -0.7$. For any $\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathcal{D}^n$, all x_i will either be -1 or 1 . Therefore, since $K(|x - (-1)|) = K(|x - 1|)$, it follows that $w_i^S(x) = \frac{1}{n}$ for all $1 \leq i \leq n$. Since $x_i = 1$ with probability 0.9, it follows that with high probability x will be classified as 1 which means that f , the output of W , is not robust at $x = -1$. Thus f has astuteness at most 0.9 which means that W is *not* r -consistent for $r = 0.3$.

1.4 General Distributions

We next consider more general data distributions, where data from different classes may be close together in space, and may even overlap. Observe that unlike the r -separated case, here there may be no classifier with astuteness one. Thus, a natural question is: what does the optimally astute classifier look like, and how can we build non-parametric classifiers to this limit?

1.4.1 The r -Optimal Classifier and Adversarial Pruning

[11] propose a large-sample limit – called the r -optimal – and show that it is analogous to the Bayes Optimal classifier for robustness. More specifically, given a data distribution D , to find the r -optimal classifier, we solve the following optimization problem.

$$\begin{aligned} \max_{S_{+1}, S_{-1}} & \int_{x \in S_{+1}} p(y = +1|x) d\mu_{\mathcal{D}}(x) + \\ & \int_{x \in S_{-1}} p(y = -1|x) d\mu_{\mathcal{D}}(x) \\ \text{subject to} & d(S_{+1}, S_{-1}) > 2r \end{aligned} \tag{1.1}$$

Then, the r -optimal classifier is defined as follows.

Definition 16. [11] Fix r, \mathcal{D} . Let S_{+1}^* and S_{-1}^* be any optimizers of (1.1). Then the r -optimal classifier, g_r^* is any classifier such that $g_r^*(x) = j$ whenever $d(S_j^*, x) \leq r$.

[11] show that the r -optimal classifier achieves the optimal astuteness – out of all classifiers on the data distribution \mathcal{D} ; hence, it is a robustness analogue to the Bayes Optimal Classifier. Therefore, for general distributions, the goal in robust classification is to find non-parametric algorithms that output classifiers that converge towards g_r^* .

To find robust classifiers, [11] propose Adversarial Pruning – a defense method that preprocesses the training data by making it better separated. More specifically, Adversarial

Pruning takes as input a training dataset S and a radius r , and finds the largest subset of the training set where differently labeled points are at least distance $2r$ apart.

Definition 17. A set $S_r \subset \mathcal{X} \times \{\pm 1\}$ is said to be *r -separated* if for all $(x_1, y_1), (x_2, y_2) \in S_r$, if $y_1 \neq y_2$, then $d(x_1, x_2) > 2r$. To *adversarially prune* a set S is to return its largest r -separated subset. We let $\text{AdvPrun}(S, r)$ denote the result of adversarially pruning S .

Once an r -separated subset S_r of the training set is found, a standard non-parametric method is trained on S_r . While [11] show good empirical performance of such algorithms, no formal guarantees are provided. We next formally characterize when adversarial pruning followed by a non-parametric method results in a classifier that is provably r -consistent.

Specifically, we consider analyzing the general algorithm provided in Algorithm 1.

Algorithm 1: RobustNonPar

- 1 **Input:** $S \sim \mathcal{D}^n$, weight function W , robustness radius r ;
 - 2 $S_r \leftarrow \text{AdvPrun}(S, r)$;
 - 3 **Output:** W_{S_r} ;
-

1.4.2 Convergence Guarantees

We begin with some notation. For any weight function W and radius $r > 0$, we let $\text{RobustNonPar}(W, r)$ represent the weight function that outputs weights for $S \sim \mathcal{D}^n$ according to $\text{RobustNonPar}(S, W, r)$. In particular, this can be used to convert any weight function algorithm into a new weight function which takes robustness into account. A natural question is, for which weight functions W is $\text{RobustNonPar}(W, r)$ r -consistent? Our next theorem provides sufficient conditions for this.

Theorem 18. Let W be a weight function over $\mathcal{X} \times \{\pm 1\}$, and let \mathcal{D} be a distribution over $\mathcal{X} \times \{\pm 1\}$. Fix $r > 0$. Let $S_r = \text{AdvPrun}(S, r)$. For convenience, relabel x_i, y_i so that $S_r =$

$\{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m)\}$. Suppose that for any $0 < a < b$,

$$\lim_{n \rightarrow \infty} \mathbb{E}_{S \sim \mathcal{D}^n} \left[\frac{1}{m} \sum_{i=1}^m \sup_{x \in B(x_i, a)} \sum_{j=1}^m w_j^{S_r}(x) I_{||x_j - x|| > b} \right] = 0.$$

Then $\text{RobustNonPar}(W, r)$ is r -consistent with respect to \mathcal{D} .

Remark:

There are two important differences between the conditions in Theorem 18 and Theorem 12.

1. We replace S with S_r .
2. The expectation over $X \sim \mathcal{D}_{\mathcal{X}}$ is replaced with an average over $\{x_1, x_2, \dots, x_m\}$. The intuition here is that we are replacing \mathcal{D} with a uniform distribution over S_r . While \mathcal{D} may not be r -separated, the uniform distribution over S_r is, and represents the region of points where our classifier is astute.

A natural question is what satisfies the conditions in Theorem 18. We next show that k_n -nearest neighbors and kernel classifiers with rapidly decaying kernel functions continue to satisfy the conditions in Theorem 18; this means that these classifiers, when combined with Adversarial Pruning, will converge to r -optimal classifiers in the large sample limit.

Corollary 19. *Let k_n be a sequence with $\lim_{n \rightarrow \infty} \frac{k_n}{n} = 0$, and let M denote the k_n -nearest neighbor algorithm. Then for any $r > 0$, $\text{RobustNonPar}(M, r)$ is r -consistent.*

Remark:

Corollary 19 gives a formal guarantee in the large sample limit for the modified nearest-neighbor algorithm proposed by [11].

Corollary 20. *Let W be a kernel classifier over $\mathcal{X} \times \{\pm 1\}$ constructed from K and h_n . Suppose the following properties hold for K and h_n .*

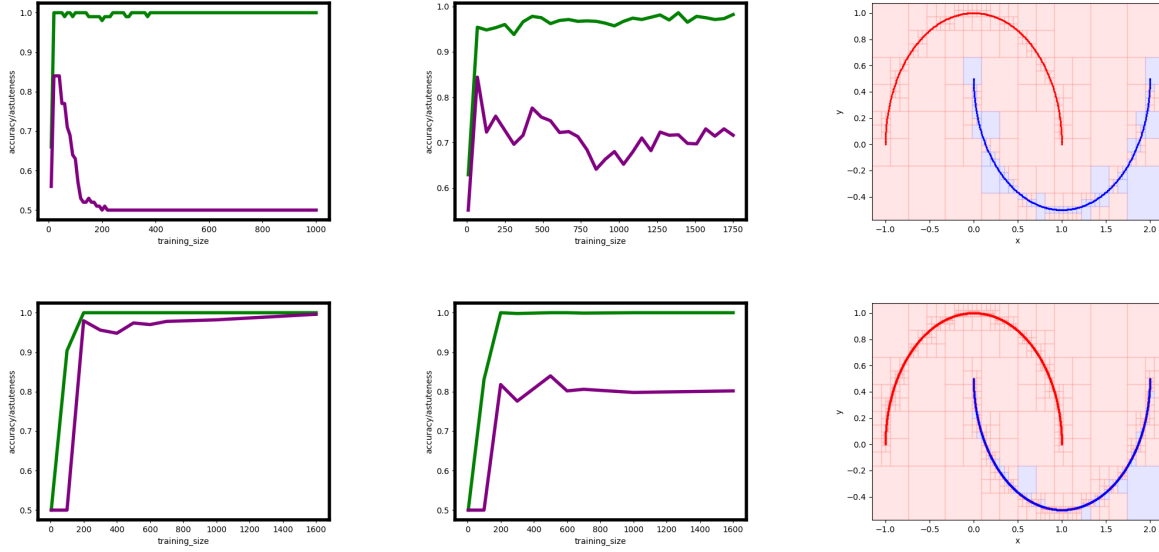


Figure 1.2. Empirical accuracy/astuteness of different classifiers as a function of training sample size. Accuracy is shown in green, astuteness in purple. Left : Noiseless Setting. Right: Noisy Setting. Top Row: Histogram Classifier, Bottom Row: 1-Nearest Neighbor

1. For any $c > 1$, $\lim_{x \rightarrow \infty} \frac{K(cx)}{K(x)} = 0$.

2. $\lim_{n \rightarrow \infty} h_n = 0$.

Then for any $r > 0$, $\text{RobustNonPar}(W, r)$ is r -consistent.

Observe again that Condition 1. is satisfied by any K that decreases more rapidly than an inverse polynomial kernel; it is thus satisfied by most popular kernels, such as the Gaussian kernel.

1.5 Validation

Our theoretical results are, by nature, large sample; we next validate how well they apply to the finite sample case by trying them out on a simple example. In particular, we ask the following question:

How does the robustness of non-parametric classifiers change with increasing sample size?

This question is considered in the context of two simple non-parametric classifiers – one nearest neighbor (which is guaranteed to be r -consistent) and histograms (which is not). To be able to measure performance with increasing data size, we look at a simple synthetic dataset – the Half Moons.

1.5.1 Experimental Setup

Classifiers and Dataset.

We consider two different classification algorithms – one nearest neighbor (NN) and a Histogram Classifier (HC). We use the Halfmoon dataset with two settings of the gaussian noise parameter σ , $\sigma = 0$ (Noiseless) and $\sigma = 0.08$ (Noisy). For the Noiseless setting, observe that the data is already 0.1-separated; for the Noisy setting, we use Adversarial Pruning (Algorithm 1) with parameter $r = 0.1$ for both classification methods.

Performance Measure.

We evaluate robustness with respect to the ℓ_∞ metric, that is commonly used in the adversarial examples literature. Specifically, for each classifier, we calculate the *empirical astuteness*, which is the fraction of test examples on which it is astute.

Observe that computing the empirical astuteness of a classifier around an input x amounts to finding the adversarial example that is *closest to x* according to the ℓ_∞ norm. For the 1-nearest neighbor, we do this using the optimal attack algorithm proposed by Yang et. al. [11]. For the histogram classifier, we use the optimal attack framework proposed by [11], and show that the structure of the classifier can be exploited to solve the convex program efficiently. Details are in Appendix C.

We use an attack radius of $r = 0.1$ for the Noiseless setting, and $r = 0.09$ for the Noisy setting. For all classification algorithms, we plot the empirical astuteness as a function of the training set size. As a baseline, we also plot their standard accuracy on the test set.

1.5.2 Results

The results are presented in Figure 1.2; the left two panels are for the Noiseless setting while the two center ones are for the Noisy setting.

The results show that as predicted by our theory, for the Noiseless setting, the empirical astuteness of nearest neighbors converges to 1 as the training set grows. For Histogram Classifiers, the astuteness converges to 0.5 – indicating that the classifier may grow less and less astute with higher sample size even for well-separated data. This is plausibly because the cell size induced by the histogram grows smaller with growing training data; thus, the classifier that outputs the default label -1 in empty cells is incorrect on adversarial examples that are close to a point with $+1$ label, but belongs to a different, empty cell. The rightmost panels in Figure 1.2 provide a visual illustration of this process.

For the Noisy setting, the empirical astuteness of adversarial pruning followed by nearest neighbors converges to 0.8. For histograms with adversarial pruning, the astuteness converges to 0.7, which is higher than the noiseless case but still clearly sub-optimal.

1.5.3 Discussion

Our results show that even though our theory is asymptotic, our predictions continue to be relevant in finite sample regimes. In particular, on well-separated data, nearest neighbors that we theoretically predict to be intrinsically robust is robust; histogram classifiers, which do not satisfy the conditions in Theorem 12 are not. Our predictions continue to hold for data that is not well-separated. Nearest neighbors coupled with Adversarial Pruning continues to be robust with growing sample size, while histograms continue to be non-robust. Thus our theory is confirmed by practice.

1.6 Conclusion

In conclusion, we rigorously analyze when non-parametric methods provide classifiers that are robust in the large sample limit. We provide a general condition that characterizes when non-parametric methods are robust on well-separated data, and show that Adversarial Pruning of [11] works on data that is not well-separated.

Our results serve to provide a set of guidelines that can be used for designing non-parametric methods that are robust and accurate on well-separated data; additionally, we demonstrate that when data is not well-separated, preprocessing by adversarial pruning [11] does lead to optimally astute solutions in the large sample limit.

Chapter 2

Consistent Non-Parametric Methods for Maximizing Robustness

2.1 Introduction

Adversarially robust classification, that has been of much recent interest, is typically formulated as follows. We are given data drawn from an underlying distribution D , a metric d , as well as a pre-specified robustness radius r . We say that a classifier c is r -robust at an input x if it predicts the same label on a ball of radius r around x . Our goal in robust classification is to find a classifier c that maximizes astuteness, which is defined as accuracy on those examples where c is also r -robust.

While this formulation has inspired a great deal of recent work, both theoretical and empirical [12, 13, 14, 15, 1, 16, 18, 19, 20, 21, 34], a major limitation is that enforcing a pre-specified robustness radius r may lead to sub-optimal accuracy *and* robustness. To see this, consider what would be an ideally robust classifier the example in Figure 2.1. For simplicity, suppose that we know the data distribution. In this case, a classifier that has an uniformly large robustness radius r will misclassify some points from the blue cluster on the left, leading to lower accuracy. This is illustrated in panel (a), in which large robustness radius leads to intersecting robustness regions. On the other hand, in panel (b), the blue cluster on the right is highly separated from the red cluster, and could be accurately classified with a high margin. But this will not happen if the robustness radius is set small enough to avoid the problems posed in

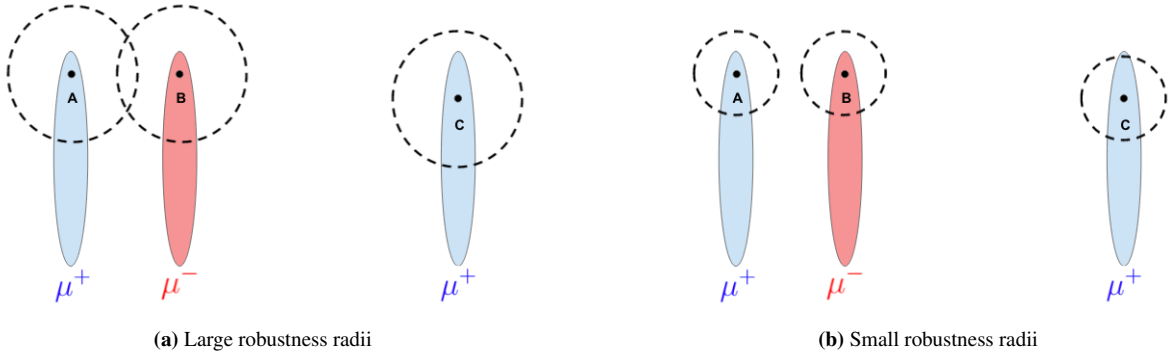


Figure 2.1. A data distribution demonstrating the difficulties with fixed radius balls for robustness regions. The red represents negatively labeled points, and the blue positive. If the robustness radius is set too large (panel (a)), then the regions of A and B intersect leading to a loss of accuracy. If the radius is set too small (panel (b)), this leads to a loss of robustness at point C where in principle it should be possible to defend against a larger amount of adversarial attacks.

panel (a). Thus, enforcing a fixed robustness radius that applies to the entire dataset may lead to lower accuracy and lower robustness.

In this work, we propose an alternative formulation of robust classification that ensures that in the large sample limit, there is no robustness-accuracy trade off, and that regions of space with higher separation are classified more robustly. An extra advantage is that our formulation is achievable by existing methods. In particular, we show that two very common non-parametric algorithms – nearest neighbors and kernel classifiers – achieve these properties in the large sample limit.

Our formulation is built on the notion of a new large-sample limit. In the standard statistical learning framework, the large-sample ideal is the Bayes optimal classifier that maximizes accuracy on the data distribution, and is undefined outside. Since this is not always robust with radius r , prior work introduces the notion of an r -optimal classifier [11] that maximizes accuracy on points where it is also r -robust. However, this classifier also suffers from the same challenges as the example in Figure 2.1.

We depart from both by introducing a new limit that we call the neighborhood preserving Bayes optimal classifier, described as follows. Given an input x that lies in the support of the data distribution D , it predicts the same label as the Bayes optimal. On an x outside the support,

it outputs the prediction of the Bayes Optimal on the nearest neighbor of x *within* the support of D . The first property ensures that there is no loss of accuracy – since it always agrees with the Bayes Optimal within the data distribution. The second ensures higher robustness in regions that are better separated. Our goal is now to design classifiers that converge to the neighborhood preserving Bayes optimal in the large sample limit; this ensures that with enough data, the classifier will have accuracy approaching that of the Bayes optimal, as well as higher robustness where possible without sacrificing accuracy.

We next investigate how to design classifiers with this convergence property. Our starting point is classical statistical theory [10] that shows that a class of methods known as weight functions will converge to a Bayes optimal in the large sample limit provided certain conditions hold; these include k -nearest neighbors under certain conditions on k and n , certain kinds of decision trees as well as kernel classifiers. Through an analysis of weight functions, we next establish precise conditions under which they converge to the neighborhood preserving Bayes optimal in the large sample limit. As expected, these are stronger than standard convergence to the Bayes optimal. In the large sample limit, we show that k_n -nearest neighbors converge to the neighborhood preserving Bayes optimal provided $k_n = \omega(\log n)$, and kernel classifiers converge to the neighborhood preserving Bayes optimal provided certain technical conditions (such as the bandwidth shrinking sufficiently slowly). By contrast, certain types of histograms do not converge to the neighborhood preserving Bayes optimal, even if they do converge to the Bayes optimal. We round these off with a lower bound that shows that for nearest neighbor, the condition that $k_n = \omega(\log n)$ is tight. In particular, for $k_n = O(\log n)$, there exist distributions for which k_n -nearest neighbors provably fails to converge towards the neighborhood preserving Bayes optimal (despite converging towards the standard Bayes optimal).

In summary, the contributions of the paper are as follows. First, we propose a new large sample limit the neighborhood preserving Bayes optimal and a new formulation for robust classification. We then establish conditions under which weight functions, a class of non-parametric methods, converge to the neighborhood preserving Bayes optimal in the large sample

limit. Using these conditions, we show that k_n -nearest neighbors satisfy these conditions when $k_n = \omega(\log n)$, and kernel classifiers satisfy these conditions provided the kernel function K has faster than polynomial decay, and the bandwidth parameter h_n decreases sufficiently slowly.

To complement these results, we also include negative examples of non-parametric classifiers that do not converge. We provide an example where histograms do not converge to the neighborhood preserving Bayes optimal with increasing n . We also show a lower bound for nearest neighbors, indicating that $k_n = \omega(\log n)$ is both necessary and sufficient for convergence towards the neighborhood preserving Bayes optimal.

Our results indicate that the neighborhood preserving Bayes optimal formulation shows promise and has some interesting theoretical properties. We leave open the question of coming up with other alternative formulations that can better balance both robustness and accuracy for all kinds of data distributions, as well as are achievable algorithmically. We believe that addressing this would greatly help address the challenges in adversarial robustness.

2.2 Preliminaries

We consider binary classification over $\mathbb{R}^d \times \{\pm 1\}$, and let ρ denote any distance metric on \mathbb{R}^d . We let μ denote the measure over \mathbb{R}^d corresponding to the probability distribution over which instances $x \in \mathbb{R}^d$ are drawn. Each instance x is then labeled as $+1$ with probability $\eta(x)$ and -1 with probability $1 - \eta(x)$. Together, μ and η comprise our data distribution $\mathcal{D} = (\mu, \eta)$ over $\mathbb{R}^d \times \{\pm 1\}$.

For comparison to the robust case, for a classifier $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ and a distribution \mathcal{D} over $\mathbb{R}^d \times \{\pm 1\}$, it will be instructive to consider its **accuracy**, denoted $A(f, \mathcal{D})$, which is defined as the fraction of examples from \mathcal{D} that f labels correctly. Accuracy is maximized by the **Bayes Optimal classifier**: which we denote by g . It can be shown that for any $x \in \text{supp}(\mu)$, $g(x) = 1$ if $\eta(x) \geq \frac{1}{2}$, and $g(x) = -1$ otherwise.

Our goal is to build classifiers $\mathbb{R}^d \rightarrow \{\pm 1\}$ that are both accurate and robust to small

perturbations. For any example x , perturbations to it are constrained to taking place in the **robustness region** of x , denoted U_x . We will let $\mathcal{U} = \{U_x : x \in \mathbb{R}^d\}$ denote the collections of all robustness regions.

We say that a classifier $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ is **robust** at x if for all $x' \in U_x$, $f(x') = f(x)$. Combining robustness and accuracy, we say that classifier is **astute** at a point x if it is both accurate and robust. Formally, we have the following definition.

Definition 21. A classifier $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ is said to be **astute** at (x, y) with respect to robustness collection \mathcal{U} if $f(x) = y$ and f is robust at x with respect to \mathcal{U} . If \mathcal{D} is a data distribution over $\mathbb{R}^d \times \{\pm 1\}$, the **astuteness** of f over \mathcal{D} with respect to \mathcal{U} , denoted $A_{\mathcal{U}}(f, \mathcal{D})$, is the fraction of examples $(x, y) \sim \mathcal{D}$ for which f is astute at (x, y) with respect to \mathcal{U} . Thus

$$A_{\mathcal{U}}(f, \mathcal{D}) = P_{(x, y) \sim \mathcal{D}}[f(x') = y, \forall x' \in U_x].$$

Non-parametric Classifiers

We now briefly review several kinds of non-parametric classifiers that we will consider throughout this paper. We begin with *weight functions*, which are a general class of non-parametric algorithms that encompass many classic algorithms, including nearest neighbors and kernel classifiers.

Weight functions are built from training sets, $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ by assigning a function $w_i^S : \mathbb{R}^d \rightarrow [0, 1]$ that essentially scores how relevant the training point (x_i, y_i) is to the example being classified. The functions w_i^S are allowed to depend on x_1, \dots, x_n but must be independent of the labels y_1, \dots, y_n . Given these functions, a point x is classified by just checking whether $\sum y_i w_i^S(x) \geq 0$ or not. If it is nonnegative, we output $+1$ and otherwise -1 . A complete description of weight functions is included in the appendix.

Next, we enumerate several common Non-parametric classifiers that can be construed as weight functions. Details can be found in the appendix.

Histogram classifiers partition the domain \mathbb{R}^d into cells recursively by splitting cells

that contain a sufficiently large number of points x_i . This corresponds to a weight function in which $w_i^S(x) = \frac{1}{k_x}$ if x_i is in the same cell as x , where k_x denotes the number of points in the cell containing x .

k_n -nearest neighbors corresponds to a weight function in which $w_i^S(x) = \frac{1}{k_n}$ if x_i is one of the k_n nearest neighbors of x , and $w_i^S(x) = 0$ otherwise.

Kernel-Similarity classifiers are weight functions built from a kernel function $K : \mathbb{R}_{\geq 0} \rightarrow \mathbb{R}_{\geq 0}$ and a window size $(h_n)_1^\infty$ such that $w_i^S(x) \propto K(\rho(x, x_i)/h_n)$ (we normalize by dividing by $\sum_1^n K(\rho(x, x_i)/h_n)$).

2.3 The Neighborhood preserving Bayes optimal classifier

Robust classification is typically studied by setting the robustness regions, $\mathcal{U} = \{U_x\}_{x \in \mathbb{R}^d}$, to be balls of radius r centered at x , $U_x = \{x' : \rho(x, x') \leq r\}$. The quantity r is the robustness radius, and is typically set by the practitioner (before any training has occurred).

This method has a limitation with regards to trade-offs between accuracy and robustness. To increase the margin or robustness, we must have a large robustness radius (thus allowing us to defend from larger adversarial attacks). However, with large robustness radii, this can come at a cost of accuracy, as it is not possible to robustly give different labels to points with intersecting robustness regions.

For an illustration, consider Figure 2.1. Here we consider a data distribution $D = (\mu, \eta)$ in which the blue regions denote all points with $\eta(x) > 0.5$ (and thus should be labeled $+$), and the red regions denote all points with $\eta(x) < 0.5$ (and thus should be labeled $-$). Observe that it is not possible to be simultaneously accurate and robust at points A, B while enforcing a large robustness radius, as demonstrated by the intersecting balls. While this can be resolved by using a smaller radius, this results in losing out on potential robustness at point C . In principal, we should be able to afford a large margin of robustness about C due to its relatively far distance from the red regions.

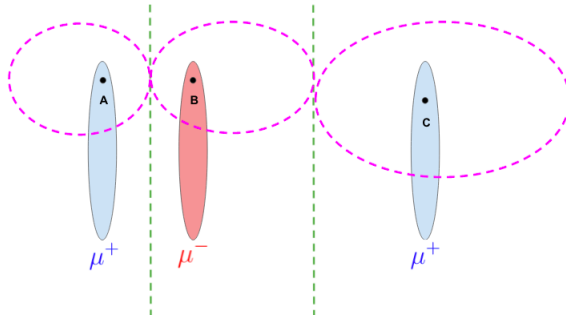


Figure 2.2. The decision boundary of the neighborhood preserving Bayes optimal classifier is shown in green, and the neighborhood preserving robust region of x is shown in pink. The former consists of points equidistant from μ^+, μ^- , and the latter consists of points equidistant from x, μ^+ .

Motivated by this issue, we seek to find a formalism for robustness that allows us to simultaneously avoid paying for any accuracy-robustness trade-offs and *adaptively* size robustness regions (thus allowing us to defend against a larger range of adversarial attacks at points that are located in more homogenous zones of the distribution support). To approach this, we will first provide an ideal limit object: a classifier that has the same accuracy as the Bayes optimal (thus meeting our first criteria) that has good robustness properties. We call this the neighborhood preserving Bayes optimal classifier, defined as follows.

Definition 22. Let $\mathcal{D} = (\mu, \eta)$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then the **neighborhood preserving Bayes optimal classifier** of \mathcal{D} , denoted $g_{neighbor}$, is the classifier defined as follows. Let $\mu^+ = \{x : \eta(x) \geq \frac{1}{2}\}$ and $\mu^- = \{x : \eta(x) < \frac{1}{2}\}$. Then for any $x \in \mathbb{R}^d$, $g_{neighbor}(x) = +1$ if $\rho(x, \mu^+) \leq \rho(x, \mu^-)$, and $g_{neighbor}(x) = -1$ otherwise.

This classifier can be thought of as the most robust classifier that matches the accuracy of the Bayes optimal. We call it *neighborhood preserving* because it extends the Bayes optimal classifier into a local neighborhood about every point in the support. For an illustration, refer to Figure 2.2, which plots the decision boundary of the neighborhood preserving Bayes optimal for an example distribution.

Next, we turn our attention towards measuring its robustness, which must be done with respect to some set of robustness regions $\mathcal{U} = \{U_x\}$. While these regions U_x can be nearly

arbitrary, we seek regions U_x such that $A_{\mathcal{U}}(g_{\max}, \mathcal{D}) = A(g_{\text{bayes}}, \mathcal{D})$ (our astuteness equals the maximum possible accuracy) and U_x are “as large as possible” (representing large robustness). To this end, we propose the following regions.

Definition 23. Let $\mathcal{D} = (\mu, \eta)$ be a data distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let $\mu^+ = \{x : \eta(x) > \frac{1}{2}\}$, $\mu^- = \{x : \eta(x) < \frac{1}{2}\}$, and $\mu^{1/2} = \{x : \eta(x) = \frac{1}{2}\}$. For $x \in \mu^+$, we define the **neighborhood preserving robustness region**, denoted V_x , as

$$V_x = \{x' : \rho(x, x') < \rho(\mu^- \cup \mu^{1/2}, x')\}.$$

It consists of all points that are closer to x than they are to $\mu^- \cup \mu^{1/2}$ (points oppositely labeled from x). We can use a similar definition for $x \in \mu^-$. Finally, if $x \in \mu^{1/2}$, we simply set $V_x = \{x\}$.

These robustness regions take advantage of the structure of the neighborhood preserving Bayes optimal. They can essentially be thought of as regions that maximally extend from any point x in the support of \mathcal{D} to the decision boundary of the neighborhood preserving Bayes optimal. We include an illustration of the regions V_x for an example distribution in Figure 2.2.

As a technical note, for $x \in \text{supp}(\mathcal{D})$ with $\eta(x) = 0.5$, we give them a trivial robustness region. The rational for doing this is that $\eta(x) = 0.5$ is an edge case that is arbitrary to classify, and consequently enforcing a robustness region at that point is arbitrary and difficult to enforce.

We now formalize the robustness and accuracy guarantees of the max-margin Bayes optimal classifier with the following two results.

Theorem 24. (Accuracy) Let \mathcal{D} be a data distribution. Let \mathcal{V} denote the collection of neighborhood preserving robustness regions, and let g denote the Bayes optimal classifier. Then the neighborhood preserving Bayes optimal classifier, g_{neighbor} , satisfies $A_{\mathcal{V}}(g_{\text{neighbor}}, \mathcal{D}) = A(g, \mathcal{D})$, where $A(g, \mathcal{D})$ denotes the accuracy of the Bayes optimal. Thus, g_{neighbor} maximizes accuracy.

Theorem 25. (Robustness) Let \mathcal{D} be a data distribution, let f be a classifier, and let \mathcal{U} be a set of robustness regions. Suppose that $A_{\mathcal{U}}(f, \mathcal{D}) = A(g, \mathcal{D})$, where g denotes the Bayes optimal

classifier. Then there exists $x \in \text{supp}(\mathcal{D})$ such that $V_x \not\subset U_x$, where V_x denotes the neighborhood preserving robustness region about x . In particular, we cannot have V_x be a strict subset of U_x for all x .

Theorem 24 shows that the neighborhood preserving Bayes classifier achieves maximal accuracy, while Theorem 25 shows that achieving a strictly higher robustness (while maintaining accuracy) is not possible; while it is possible to make accurate classifiers which have higher robustness than g_{neighbor} in some regions of space, it is not possible for this to hold across all regions. Thus, the neighborhood preserving Bayes optimal classifier can be thought of as a local maximum to the constrained optimization problem of maximizing robustness subject to having maximum (equal to the Bayes optimal) accuracy.

2.3.1 Neighborhood Consistency

Having defined the neighborhood preserving Bayes optimal classifier, we now turn our attention towards building classifiers that converge towards it. Before doing this, we must precisely define what it means to converge. Intuitively, this consists of building classifiers whose robustness regions “approach” the robustness regions of the neighborhood preserving Bayes optimal classifier. This motivates the definition of *partial neighborhood preserving robustness regions*.

Definition 26. Let $0 < \kappa < 1$ be a real number, and let $\mathcal{D} = (\mu, \eta)$ be a data distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let $\mu^+ = \{x : \eta(x) > \frac{1}{2}\}$, $\mu^- = \{x : \eta(x) < \frac{1}{2}\}$, and $\mu^{1/2} = \{x : \eta(x) = \frac{1}{2}\}$. For $x \in \mu^+$, we define the **neighborhood preserving robustness region**, denoted V_x , as

$$V_x = \{x' : \rho(x, x') < \kappa \rho(\mu^- \cup \mu^{1/2}, x')\}.$$

It consists of all points that are closer to x than they are to $\mu^- \cup \mu^{1/2}$ (points oppositely labeled from x) by a factor of κ . We can use a similar definition for $x \in \mu^-$. Finally, if $\eta(x) = \frac{1}{2}$, we simply set $V_x^\kappa = \{x\}$.

Observe that $V_x^\kappa \subset V_x$ for all $0 < \kappa < 1$, and thus being robust with respect to V_x^κ is a milder condition than V_x . Using this notion, we can now define margin consistency.

Definition 27. A learning algorithm A is said to be **neighborhood consistent** if the following holds for any data distribution $\mathcal{D} = (\mu, \eta)$ where η is continuous on its support. For any $0 < \varepsilon, \delta, \kappa < 1$, there exists N such that for all $n \geq N$, with probability at least $1 - \delta$ over $S \sim \mathcal{D}^n$,

$$A_{\mathcal{V}^\kappa}(A_S, D) \geq A(g, \mathcal{D}) - \varepsilon,$$

where g denotes the Bayes optimal classifier and A_S denotes the classifier learned by algorithm A from dataset S .

This condition essentially says that the astuteness of the classifier learned by the algorithm converges towards the accuracy of the Bayes optimal classifier. Furthermore, we stipulate that this holds as long as the astuteness is measured with respect to some \mathcal{V}^κ . Observe that as $\kappa \rightarrow 1$, these regions converge towards the neighborhood preserving robustness regions, thus giving us a classifier with robustness effectively equal to that of the neighborhood preserving Bayes optimal classifier.

2.4 Neighborhood Consistent Non-Parametric Classifiers

Having defined neighborhood consistency, we turn to the following question: which non-parametric algorithms are neighborhood consistent? Our starting point will be the standard literature for the convergence of non-parametric classifiers with regard to accuracy. We begin by considering the standard conditions for k_n -nearest neighbors to converge (in accuracy) towards the Bayes optimal.

k_n -nearest neighbors is *consistent* if and only if the following two conditions are met: $\lim_{n \rightarrow \infty} k_n = \infty$, and $\lim_{n \rightarrow \infty} \frac{k_n}{n} = 0$. The first condition guarantees that each point is classified by using an increasing number of nearest neighbors (thus making the probability of a misclassification small), and the second condition guarantees that each point is classified using only points

very close to it. We will refer to the first condition as *precision*, and the second condition as *locality*. A natural question is whether the same principles suffice for neighborhood consistency as well. We began by showing that without any additional constraints, the answer is no.

Theorem 28. *Let $\mathcal{D} = (\mu, \eta)$ be the data distribution where μ denotes the uniform distribution over $[0, 1]$ and η is defined as: $\eta(x) = x$. Over this space, let ρ be the euclidean distance metric. Suppose $k_n = O(\log n)$ for $1 \leq n < \infty$. Then k_n -nearest neighbors is not neighborhood consistent with respect to \mathcal{D} .*

The issue in the example above is that for smaller k_n , k_n -nearest neighbors lacks sufficient precision. For neighborhood consistency, points must be labeled using even more training points than are needed accuracy. This is because the classifier must be uniformly correct across the entirety of V_x^K . Thus, to build neighborhood consistent classifiers, we must bolster the precision from the standard amount used for standard consistency. To do this, we begin by introducing *splitting numbers*, a useful tool for bolstering the precision of weight functions.

2.4.1 Splitting Numbers

We will now generalize beyond nearest neighbors to consider weight functions. Doing so will allow us to simultaneously analyze nearest neighbors and kernel classifiers. To do so, we must first rigorously substantiate our intuitions about increasing precision into concrete requirements. This will require several technical definitions.

Definition 29. *Let μ be a probability measure over \mathbb{R}^d . For any $x \in \mathbb{R}^d$, the **probability radius** $r_p(x)$ is the smallest radius for which $B(x, r_p(x))$ has probability mass at least p . More precisely, $r_p(x) = \inf\{r : \mu(B(x, r)) \geq p\}$.*

Definition 30. *Let W be a weight function and let $S = \{x_1, x_2, \dots, x_n\}$ be any finite subset of \mathbb{R}^d . For any $x \in \mathbb{R}^d$, $\alpha \geq 0$, and $0 \leq \beta \leq 1$, let $W_{x, \alpha, \beta} = \{i : \rho(x, x_i) \leq \alpha, w_i^S(x) \geq \beta\}$. Then the **splitting number** of W with respect to S , denoted as $T(W, S)$ is the number of distinct subsets*

generated by $W_{x,\alpha\beta}$ as x ranges over \mathbb{R}^d , α ranges over $[0, \infty)$, and β ranges over $[0, 1]$. Thus $T(W, S) = |\{W_{x,\alpha,\beta} : x \in \mathbb{R}^d, 0 \leq \alpha, 0 \leq \beta \leq 1\}|$.

Splitting numbers allow us to ensure high amounts of precision over a weight function. To prove neighborhood consistency, it is necessary for a classifier to be correct at *all* points in a given region. Consequently, techniques that consider a single point will be insufficient. The splitting number provides a mechanism for studying entire regions simultaneously. For more details on splitting numbers, we include several examples in the appendix.

2.4.2 Sufficient Conditions for Neighborhood Consistency

We now state our main result.

Theorem 31. *Let W be a weight function, \mathcal{D} a distribution over $\mathbb{R}^d \times \{\pm 1\}$, \mathcal{U} a neighborhood preserving collection, and $(t_n)_1^\infty$ be a sequence of positive integers such that the following four conditions hold.*

1. *W is consistent (with resp. to accuracy) with resp. to \mathcal{D} .*
2. *For any $0 < p < 1$, $\lim_{n \rightarrow \infty} E_{S \sim \mathcal{D}^n} [\sup_{x \in \mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x, x_i) > r_p(x)}] = 0$.*
3. *$\lim_{n \rightarrow \infty} E_{S \sim \mathcal{D}^n} [t_n \sup_{x \in \mathbb{R}^d} w_i^S(x)] = 0$.*
4. *$\lim_{n \rightarrow \infty} E_{S \sim \mathcal{D}^n} \frac{\log T(W, S)}{t_n} = 0$.*

Then W is neighborhood consistent with respect to \mathcal{D} .

Remarks: Condition 1 is necessary because neighborhood consistency implies standard consistency – or, convergence in accuracy to the Bayes Optimal. Standard consistency has been well studied for non-parametric classifiers, and there are a variety of results that can be used to ensure it – for example, Stone’s Theorem (included in the appendix).

Conditions 2. and 3. are stronger version of conditions 2. and 3. of Stone’s theorem. In particular, both include a supremum taken over all $x \in \mathbb{R}^d$ as opposed to simply considering a random point $x \sim \mathcal{D}$. This is necessary for ensuring correct labels on entire regions of points simultaneously. We also note that the dependence on $r_p(x)$ (as opposed to some fixed r) is a key

property used for adaptive robustness. This allows the algorithm to adjust to potential differing distance scales over different regions in \mathbb{R}^d . This idea is reminiscent of the analysis given in [35], which also considers probability radii.

Condition 4. is an entirely new condition which allows us to simultaneously consider all $T(W, S)$ subsets of S . This is needed for analyzing weighted sums with arbitrary weights.

Next, we apply Theorem 31 to get specific examples of margin consistent non-parametric algorithms.

2.4.3 Nearest Neighbors and Kernel Classifiers

We now provide sufficient conditions for k_n -nearest neighbors to be neighborhood consistent.

Corollary 32. *Suppose $(k_n)_1^\infty$ satisfies (1) $\lim_{n \rightarrow \infty} \frac{k_n}{n} = 0$, and (2) $\lim_{n \rightarrow \infty} \frac{\log n}{k_n} = 0$. Then k_n -nearest neighbors is neighborhood consistent.*

As a result of Theorem 28, corollary 32 is tight for nearest neighbors. Thus k_n nearest neighbors is neighborhood consistent if and only if $k_n = \omega(\log n)$.

Next, we give sufficient conditions for a kernel-similarity classifier.

Corollary 33. *Let W be a kernel classifier over $\mathbb{R}^d \times \{\pm 1\}$ constructed from $K : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ and h_n . Suppose the following properties hold.*

1. *K is decreasing, and satisfies $\int_{\mathbb{R}^d} K(\|x\|) dx < \infty$.*
2. *$\lim_{n \rightarrow \infty} h_n = 0$ and $\lim_{n \rightarrow \infty} n h_n^d = \infty$.*
3. *For any $c > 1$, $\lim_{x \rightarrow \infty} \frac{K(cx)}{K(x)} = 0$.*
4. *For any $x \geq 0$, $\lim_{n \rightarrow \infty} \frac{n}{\log n} K(\frac{x}{h_n}) = \infty$.*

Then W is neighborhood consistent.

Observe that conditions 1. 2. and 3. are satisfied by many common Kernel functions such as the Gaussian or Exponential kernel ($K(x) = \exp(-x^2)/K(x) = \exp(-x)$). Condition 4. can be

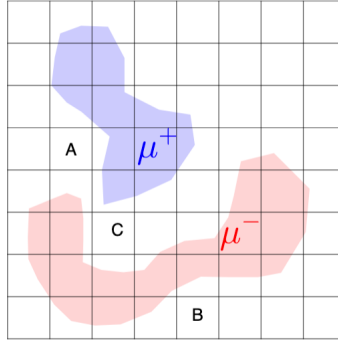


Figure 2.3. we have a histogram classifier being applied to the blue and red regions. The classifier will be unable to construct good labels in the cells labeled A, B, C , and consequently will not be robust with respect to V_x^K for sufficiently large κ .

similarly satisfied by just increasing h_n to be sufficiently large. Overall, this theorem states that Kernel classification is neighborhood consistent as long as the bandwidth shrinks slowly enough.

2.4.4 Histogram Classifiers

Having discussed neighborhood consistent nearest-neighbors and kernel classifier, we now turn our attention towards another popular weight function, histogram classifiers. Recall that histogram classifiers operate by partitioning their input space into increasingly small cells, and then classifying each cell by using a majority vote from the training examples within that cell (a detailed description can be found in the appendix). We seek to answer the following question: is increasing precision sufficient for making histogram classifiers neighborhood consistent? Unfortunately, the answer this turns out not to be no. The main issue is that histogram classifiers have no mechanism for performing classification outside the support of the data distribution.

For an example of this, refer to Figure 2.3. Here we see a distribution being classified by a histogram classifier. Observe that the cell labeled A contains points that are strictly closer to μ^+ than μ^- , and consequently, for sufficiently large κ , V_x^K will intersect A for some point $x \in \mu^+$. A similar argument holds for the cells labeled B and C . However, since A, B, C are all in cells that will never contain any data, they will never be labeled in a meaningful way. Because of this, histogram classifiers are not neighborhood consistent.

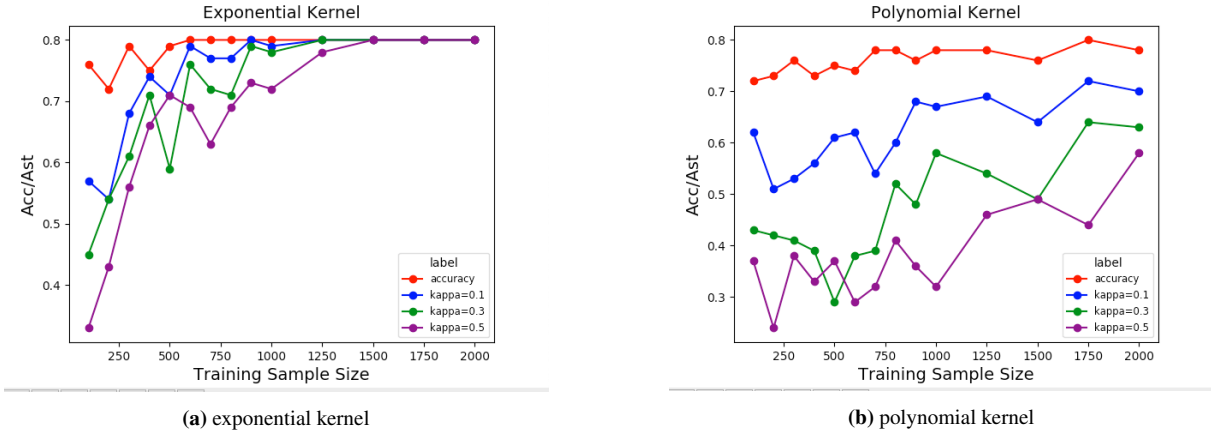


Figure 2.4. Plots of astuteness against the training sample size. In both panels, accuracy is plotted in red, and the varying levels of robustness regions ($\kappa = 0.1, 0.3, 0.5$) are given in blue, green and purple. In panel (a), observe that as sample size increases, every measure of astuteness converges towards 0.8 which is as predicted by Corollary 33. In panel (b), although the accuracy appears to converge, none of the robustness measures. In fact, they get progressively worse the larger κ gets.

2.5 Validation

To complement our theoretical large sample results for non-parametric classifiers, we now include several experiments to understand their behavior for finite samples. We seek to understand how quickly non-parametric classifiers converge towards the neighborhood preserving Bayes optimal.

We focus our attention on kernel classifiers and use two different kernel similarity functions: the first, an exponential kernel, and the second, a polynomial kernel. These classifiers were chosen so that the former meets the conditions of Corollary 33, and the latter does not. Full details on these classifiers can be found in the appendix.

To be able to measure performance with increasing data size, we look at a simple synthetic dataset over overlaid circles (see Figure ?? for an illustration) with support designed so that the data is intrinsically multiscaled. In particular, this calls for different levels of robustness in different regions. For simplicity, we use a global label noise parameter of 0.2, meaning that any sample drawn from this distribution is labeled differently than its support with probability 0.2.

Further details about our dataset are given in section ??.

Performance Measure. For a given classifier, we evaluate its astuteness at a test point x with respect to the robustness region V_x^κ (Definition 26). While these regions are not computable in practice due to their dependency on the support of the data distribution, we are able to approximate them for this synthetic example due to our explicit knowledge of the data distribution. Details for doing this can be found in the appendix. To compute the empirical astuteness of a kernel classifier W_K about test point x , we perform a grid search over all points in V_x^κ to ensure that all points in the robustness region are labeled correctly.

For each classifier, we measure the empirical astuteness by using three trials of 20 test points and taking the average. While this is a relatively small amount of test data, it suffices as our purpose is to just verify that the algorithm roughly converges towards the optimal possible astuteness. Recall that for any neighborhood consistent algorithm, as $n \rightarrow \infty$, A_{γ^κ} should converge towards A^* , the accuracy of the Bayes optimal classifier, for *any* $0 < \kappa < 1$. Thus, to verify this holds, we use $\kappa = 0.1, 0.3, 0.5$. For each of these values, we plot the empirical astuteness as the training sample size n gets larger and larger. As a baseline, we also plot their standard accuracy on the test set.

Results and Discussion: The results are presented in Figure 2.4; the left panel is for the exponential kernel, while the right one is for the polynomial kernel. As predicted by our theory, we see that in all cases, the exponential kernel converges towards the maximum astuteness regardless of the value of κ : the only difference is that the rate of convergence is slower for larger values of κ . This is, of course, expected because larger values of κ entail larger robustness regions.

By contrast, the polynomial kernel performs progressively worse for larger values of κ . This kernel was selected specifically to violate the conditions of Corollary 33, and in particular fails criteria 3. However, note that the polynomial kernel nevertheless performs well with respect to accuracy thus giving another example demonstrating the added difficulty of neighborhood consistency.

Our results bridge the gap between our asymptotic theoretical results and finite sample regimes. In particular, we see that kernel classifiers that meet the conditions of Corollary 33 are able to converge in astuteness towards the neighborhood preserving Bayes optimal classifier, while classifiers that do not meet these conditions fail.

2.6 Related Work

There is a wealth of literature on robust classification, most of which impose the same robustness radius r on the entire data. [12, 13, 14, 15, 1, 16, 17, 18, 19, 20, 21], among others, focus primarily on neural networks, and robustness regions that are ℓ_1, ℓ_2 , or ℓ_∞ norm balls of a given radius r .

[36] and [37] show how to train neural networks with different robustness radii at different points by trading off robustness and accuracy; their work differ from ours in that they focus on neural networks, their robustness regions are still norm balls, and that their work is largely empirical.

Our framework is also related to large margin classification – in the sense that the robustness regions \mathcal{U} induce a *margin constraint* on the decision boundary. The most popular large margin classifier is the Support Vector Machine[38, 39, 40] – a large margin linear classifier that minimizes the worst-case margin over the training data. Similar ideas have also been used to design classifiers that are more flexible than linear; for example, [41] shows how to build large margin Lipschitz classifiers by rounding globally Lipschitz functions. Finally, there has also been purely empirical work on achieving large margins for more complex classifiers – such as [42] for deep neural networks that minimizes the worst case margin, and [43] for metric learning to find large margin nearest neighbors. Our work differs from these in that our goal is to ensure a high enough local margin at each x , (by considering the neighborhood preserving regions V_x) as opposed to optimizing a global margin.

Finally, our analysis builds on prior work on robust classification for non-parametric

methods in the standard framework. [22, 23, 5, 11] provide adversarial attacks on non-parametric methods. Wang et. al. [5] develops a defense for 1-NN that removes a subset of the training set to ensure higher robustness. Yang et. al [11] proposes the r -optimal classifier – which is the maximally astute classifier in the standard robustness framework – and proposes a defense called Adversarial Pruning.

Theoretically, [44] provide conditions under which weight functions converge towards the r -optimal classifier in the large sample limit. They show that for r -separated distributions, where points from different classes are at least distance $2r$ or more apart, nearest neighbors and kernel classifiers satisfy these conditions. In the more general case, they use Adversarial Pruning as a preprocessing step to ensure that the training data is r -separated, and show that this preprocessing step followed by nearest neighbors or kernel classifiers leads to solutions that are robust and accurate in the large sample limit. Our result fundamentally differs from theirs in that we analyze a different algorithm, and our proof techniques are quite different. In particular, the fundamental differences between the r -optimal classifier and the neighborhood preserving Bayes optimal classifier call for different algorithms and different analysis techniques.

In concurrent work, [45] proposes a similar limit to the neighborhood preserving Bayes optimal which they refer to as the margin canonical Bayes. However, their work then focuses on a data augmentation technique that leads to convergence whereas we focus on proving the neighborhood consistency of classical non-parametric classifiers.

Chapter 3

Sample Complexity of Robust Linear Classification on Separated Data

3.1 Introduction

Motivated by the use of machine learning in safety-critical settings, adversarially robust classification has been of much recent interest. Formally, the problem is as follows. A learner is given training data drawn from an underlying distribution D , a hypothesis class \mathcal{H} , a robustness metric d , and a radius r . The learner’s goal is to find a classifier $h \in \mathcal{H}$ which has the lowest robust loss at radius r . The robust loss of a classifier is the expected fraction of examples where either $f(x) \neq y$ or where there exists an x' at distance $d(x, x') \leq r$ such that $f(x) \neq f(x')$. Robust classification thus aims to find a classifier that maximizes accuracy on examples that are distance r or more from the decision boundary, where distances are measured according to the metric d .

In this work, we ask: how many samples are needed to learn a classifier with low robust loss when \mathcal{H} is the class of linear classifiers, and d is an ℓ_p -metric? Prior work has provided both upper [46, 47] as well as lower bounds [18, 47] on the sample complexity of the problem. However, almost all look at settings where the data distribution itself is not separated – data from different classes overlap or are close together in space. In this case, the classifier that minimizes robust loss is quite different from the one that minimizes error, which often leads to strong sample complexity gaps. Many real tasks where robust solutions are desired however tend to involve well-separated data [34], and hence it is instructive to look at what happens in these

cases.

With this motivation, we consider in this work robust classification of data that is linearly r -separable. Specifically, there exists a linear classifier which has zero robust loss at robustness radius r . This case is thus the analog of the realizable case for robust classification, and we consider both upper and lower bounds in this setting.

For lower bounds, prior work [48] shows that both standard and robust linear classification have VC-dimension $O(d)$, and consequently have similar bounds on the expected loss in the worst case. However, these results do not apply to this setting since we are specifically considering well-separated data, which greatly restricts the set of possible worst-case distributions. For our lower bound, we provide a family of distributions that are linearly r -separable and where the maximum margin classifier, given n independent samples, has error $O(1/n)$. In contrast, any algorithm for finding the minimum robust loss classifier has robust loss at least $\Omega(d/n)$, where d is the data dimension. These bounds hold for all ℓ_p -norms provided $p > 1$, including $p = 2$ and $p = \infty$. Unlike prior work, our bounds do not rely on the difference in loss between the solutions with optimal robust loss and error, and hence cannot be obtained by prior techniques. Instead, we introduce a new geometric construction that exploits the fact that learning a classifier with low robust loss when data is linearly r -separated requires seeing a certain number of samples close to the margin.

For upper bounds, prior work [46] provides a bound on the Rademacher complexity of adversarially robust learning, and show that it can be worse than the standard Rademacher complexity by a factor of $d^{1/q}$ for ℓ_p -norm robustness where $1/p + 1/q = 1$. Thus, an interesting question is whether dimension-independent bounds, such as those for the accuracy under large margin classification, can be obtained for robust classification as well. Perhaps surprisingly, we show that when data is really well-separated, the answer is yes. Specifically, if the data distribution is linearly $r + \gamma$ -separable, then there exists an algorithm that will find a classifier with robust loss $O(\Delta^2/\gamma^2 n)$ at radius r where Δ is the diameter of the instance space. Observe that much like the usual sample complexity results on SVM and perceptron, this upper bound

is independent of the data dimension and depends only on the excess margin (over r). This establishes that when data is really well-separated, finding robust linear classifiers does not require a very large number of samples.

While the main focus of this work is on linear classifiers, we also show how to generalize our upper bounds to Kernel Classification, where we find a similar dynamic with the loss being governed by the excess margin in the embedded kernel space. However, we defer a thorough investigation of robust kernel classification as an avenue for future work.

Our results imply that while adversarially robust classification may be more challenging than simply accurate classification when the classes overlap, the story is different when data is well-separated. Specifically, when data is linearly (exactly) r -separable, finding an r -separated solution to robust loss ε may require $\Omega(d/\varepsilon)$ samples for some distribution families where finding an accurate solution is easier. Thus in this case, there is a gap between the sample complexities of robust and simply accurate solutions, and this is true regardless of the ℓ_p norm in which robustness is measured. In contrast, if data is even more separated – linearly $r + \gamma$ -separable – then we can obtain a dimension-independent upper bound on the sample complexity, much like the sample complexity of SVMs and perceptron. Thus, how separable the data is matters for adversarially robust classification, and future works in the area should consider separability while discussing the sample complexity.

3.1.1 Related Work

There is a large body of work [12, 13, 14, 15, 1, 16, 17, 19, 20, 21] empirically studying adversarial examples primarily in the context of neural networks. Several works [18, 49, 50] have empirically investigated trade-offs between robust and standard classification.

On the theoretical side, this phenomenon has been studied in both the parametric and non-parametric settings. On the parametric side, several works [51, 52, 6, 46, 53] have focused on finding distribution agnostic bounds of the sample complexity for robust classification. In [6], Srebro et. al. showed through an example that the VC dimension of robust learning may be

much larger than standard or accurate learning indicating that the sample complexity bounds may be higher. However, their example did not apply to linear classifiers.

[54] considers learning linear classifiers robustly, but is primarily focused on computational complexity as opposed to sample complexity.

In [46], Bartlett et. al. investigated the Rademacher complexity of robustly learning linear classifiers as well as neural networks. They showed that in both cases, the robust Rademacher complexity can be bounded in terms of the dimension of the input space – thus indicating a possible gap between standard and robust learning. However, as with the works considering VC dimension, this work is fundamentally focused on upper bounds – they do not show true lower bounds on data requirements.

Because of its simplicity and elegance, the case where the data distribution is a mixture of Gaussians has been particularly well-studied. The first such work was [18], in which Schmidt et. al. showed an $\Omega(\sqrt{d})$ gap between the standard and robust sample complexity for a mixture of two Gaussians using the ℓ_∞ norm. This was subsequently expanded upon in [55], [56] and [47]. [55] introduces a notion of “optimal transport,” which they subsequently apply to the Gaussian case, deriving a closed form expression for the optimally robust linear classifier. Their results apply to any ℓ_p norm. [56] applies expands upon [18] by consider mixtures of three Gaussians in both the ℓ_2 and ℓ_∞ norms. Finally, [47] fully generalizes the results of [18] providing tight upper and lower bounds on the standard and robust sample complexities of a mixture of two Gaussians, in any norm (including ℓ_p for $p \in [1, \infty]$). [18] and [47] bear the most relevance with our work, and we consequently carefully compare our results in section ??.

Another approach for lower and upper bounds on sample complexities for linear classifiers can be found in [48], which examines the robust VC dimension of learning linear classifiers. They show that the VC dimension is $d + 1$, just as it is in the standard case. This implies that the bounds in the robust case match the bounds in the standard case and in particular shows a lower bound of $\Omega(d/n)$ on the expected loss of learning a robust linear classifier from n samples.

While this result appears to match our lower bound, there is a crucial distinction between

the bounds. Our bound implies that there exists some distribution with a large ℓ_2 margin for which the expected robust loss must be $\Omega(d/n)$. On the other hand, standard results about learning linear classifiers on large margin data implies that the expected standard loss will be $O(1/n)$ (when running the max-margin algorithm). For this reason, our paper provides a case in the well-separated setting in which learning linear classifiers is provably more difficult (in terms of sample complexity) in the robust setting than in the standard setting. By contrast, [48] does not show this. Their paper only implies (through standard VC constructions) the existence of *some* distribution that is difficult to learn, and the standard PAC bounds cannot ensure that such a distribution also has a large ℓ_2 margin.

In the non-parametric setting, there are several works which contrast standard learning with robust learning. [5] considers the nearest neighbors algorithm, and shows how to adapt it for converging towards a robust classifier. In [11], Yang et. al. propose the *r-optimal classifier*, which is the robust analog of the Bayes optimal classifier. Through several examples they show that it is often a fundamentally different classifier - which can lead to different convergence behavior in the standard and robust settings. [44] unified these approaches by specifying conditions under which non-parametric algorithms can be adapted to converge towards the *r-optimal classifier*, thus introducing *r-consistency*, the robust analog of consistency.

3.2 Preliminaries

We consider binary classification over $\mathbb{R}^d \times \{\pm 1\}$. Our metric of choice is the ℓ_p norm, where $p > 1$ (including $p = \infty$) is arbitrary. For $x \in \mathbb{R}^d$, we will use $\|x\|_p$ to denote the ℓ_p norm of x , and consequently will use $\|x - y\|_p$ to denote the ℓ_p distance between x and y . We will also let ℓ_q denote the dual norm to ℓ_p - that is, $\frac{1}{q} + \frac{1}{p} = 1$.

We use $B_p(x, r)$ to denote the closed ℓ_p ball with center x and radius r . For any $S \subset \mathbb{R}^d$, we let $\text{diam}_p(S)$ denote its diameter: that is, $\text{diam}_p(S) = \sup_{x, y \in S} \|x - y\|_p$.

3.2.1 Standard and Robust Loss

In classical statistical learning, the goal is to learn an accurate classifier, which is defined as follows:

Definition 34. Let \mathcal{D} be a distribution over $\mathbb{R}^d \times \{\pm 1\}$, and let $f \in \{\pm 1\}^{\mathbb{R}^d}$ be a classifier. Then the **standard loss** of f over \mathcal{D} , denoted $\mathcal{L}(f, \mathcal{D})$, is the fraction of examples $(x, y) \sim \mathcal{D}$ for which f is not accurate. Thus

$$\mathcal{L}(f, \mathcal{D}) = P_{(x,y) \sim \mathcal{D}}[f(x) \neq y].$$

Next, we define robustness, and the corresponding robust loss.

Definition 35. A classifier $f \in \{\pm 1\}^{\mathbb{R}^d}$ is said to be **robust** at x with radius r if $f(x) = f(x')$ for all $x' \in B_p(x, r)$.

Definition 36. The **robust loss** of f over \mathcal{D} , denoted $\mathcal{L}_r(f, \mathcal{D})$, is the fraction of examples $(x, y) \sim \mathcal{D}$ for which f is either inaccurate at (x, y) , or f is not robust at (x, y) with radius r . Observe that this occurs if and only if there is some $x' \in B_p(x, r)$ such that $f(x') \neq y$. Thus

$$\mathcal{L}_r(f, \mathcal{D}) = P_{(x,y) \sim \mathcal{D}}[\exists x' \in B_p(x, r) \text{ s.t. } f(x') \neq y].$$

3.2.2 Expected Loss and Sample Complexity

The most common way to characterize the performance of a learning algorithm is through an (ε, δ) guarantee, which computes ε_n, δ_n such that an algorithm trained over n samples has loss at most ε_n with probability at least $1 - \delta_n$.

In this work, we use the simpler notion of *expected loss*, which is defined as follows:

Definition 37. Let A be a learning algorithm and let \mathcal{D} be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. For any $S \sim \mathcal{D}^n$, we let A_S denote the classifier learned by A from training data S . Then the **expected**

standard loss of A with respect to \mathcal{D} , denoted $EL^n(A, \mathcal{D})$ where n is the number of training samples, is defined as

$$EL^n(A, \mathcal{D}) = \mathbb{E}_{S \sim \mathcal{D}^n} \mathcal{L}(A_S, \mathcal{D}).$$

Similarly, we define the **expected robust loss** of A with respect to \mathcal{D} as

$$EL_r^n(A, \mathcal{D}) = \mathbb{E}_{S \sim \mathcal{D}^n} \mathcal{L}_r(A_S, \mathcal{D}).$$

Our main motivation for using this criteria is simplicity. Our primary goal is to compare and contrast the performances of algorithms in the standard and robust cases, and this contrast clearest when the performances are summarized as a single number (namely the expected loss) rather than an (ϵ, δ) pair.

Next, we address the notion of sample complexity. As above, sample complexity is typically defined as the minimum number of samples needed to guarantee (ϵ, δ) performance. In this work, we will instead define it solely with respect to ϵ , the expected loss.

Definition 38. Let \mathcal{D} be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ and A be a learning algorithm. Then the **standard sample complexity** of A with respect to \mathcal{D} , denoted $m^\epsilon(A, \mathcal{D})$, is the minimum number of training samples needed such that A has expected standard loss at most ϵ . Formally,

$$m^\epsilon(A, \mathcal{D}) = \min(\{n : EL^n(A, \mathcal{D}) \leq \epsilon\}).$$

Similarly, we can define the **robust sample complexity** as

$$m_r^\epsilon(A, \mathcal{D}) = \min(\{n : EL_r^n(A, \mathcal{D}) \leq \epsilon\}).$$

Bibliography

- [1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.
- [2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, March 20 2014. URL <http://arxiv.org/abs/1412.6572>.
- [3] Daniel Lowd and Christopher Meek. Adversarial learning. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 641–647. ACM, 2005. ISBN 1-59593-135-X.
- [4] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31*, pages 5014–5026. Curran Associates, Inc., 2018.
- [5] Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5120–5129, 2018.
- [6] Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, pages 2512–2530, 2019.
- [7] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya P. Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *CoRR*, abs/1906.04584, 2019. URL <http://arxiv.org/abs/1906.04584>.
- [8] Dmitrii Avdiukhin, Slobodan Mitrovic, Grigory Yaroslavtsev, and Samson Zhou. Adversarially robust submodular maximization under knapsack constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.*, pages 148–156, 2019. doi:

10.1145/3292500.3330911.

- [9] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, 1996.
- [10] Charles Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–645, 1977.
- [11] Yao-Yuan Yang, Cyrus Rashtchian, Yizhen Wang, and Kamalika Chaudhuri. Adversarial examples for non-parametric methods: Attacks, defenses and large sample limits. *CoRR*, abs/1906.03310, 2019. URL <http://arxiv.org/abs/1906.03310>.
- [12] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017.
- [13] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [14] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *ASIACCS*, 2017.
- [15] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387, 2016.
- [16] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2266–2276. Curran Associates, Inc., 2017.
- [17] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Towards proving the adversarial robustness of deep neural networks. In *Proceedings First Workshop on Formal Verification of Autonomous Vehicles, FVAV@iFM 2017, Turin, Italy, 19th September 2017.*, pages 19–26, 2017.
- [18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

- [19] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597, 2016.
- [20] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [21] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.
- [22] Laurent Amsaleg, James Bailey, Dominique Barbe, Sarah M. Erfani, Michael E. Houle, Vinh Nguyen, and Milos Radovanovic. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *2017 IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 4-7, 2017*, pages 1–6, 2017.
- [23] Chawin Sitawarin and David A. Wagner. On the robustness of deep k-nearest neighbors. In *2019 IEEE Security and Privacy Workshops, SP Workshops 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1–7, 2019.
- [24] Maksym Andriushchenko and Matthias Hein. Provably robust boosted decision stumps and trees against adversarial attacks. In H. Wallach, H. Larochelle, A. Beygelzimer, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12997–13008. Curran Associates, Inc., 2019.
- [25] Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and hardening of tree ensemble classifiers. *CoRR*, abs/1509.07892, 2015. URL <http://arxiv.org/abs/1509.07892>.
- [26] Hongge Chen, Huan Zhang, Duane S. Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. *CoRR*, abs/1902.10660, 2019.
- [27] Geoffrey W. Gates. The reduced nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 18(3):431–433, 1972.
- [28] Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 370–378, 2014.
- [29] Peter E. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 14(3):515–516, 1968.

- [30] Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1573–1583, 2017.
- [31] Aryeh Kontorovich and Roi Weiss. A bayes consistent 1-nn classifier. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, 2015.
- [32] Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Universal bayes consistency in metric spaces. *CoRR*, abs/1906.09855, 2019.
- [33] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Information Theory*, 51(1):128–142, 2005.
- [34] Yao-Yuan Yang, Cyrus Rashtchian, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at robustness vs. accuracy. In *Neural Information Processing Systems (NeurIPS)*, 2020.
- [35] Kamalika Chaudhuri and Sanjoy Dasgupta. Rates of convergence for nearest neighbor classification. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3437–3445. Curran Associates, Inc., 2014.
- [36] Minhao Cheng, Qi Lei, Pin-Yu Chen, Inderjit S. Dhillon, and Cho-Jui Hsieh. CAT: customized adversarial training for improved robustness. *CoRR*, abs/2002.06789, 2020. URL <https://arxiv.org/abs/2002.06789>.
- [37] Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. MMA training: Direct input space margin maximization through adversarial training. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [38] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3): 273–297, 1995.
- [39] Kristin P. Bennett and Erin J. Breidensteiner. Duality and geometry in SVM classifiers. In Pat Langley, editor, *Proceedings of the Seventeenth International Conference on Machine Learning (ICML 2000), Stanford University, Stanford, CA, USA, June 29 - July 2, 2000*, pages 57–64. Morgan Kaufmann, 2000.
- [40] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

- [41] Ulrike von Luxburg and Olivier Bousquet. Distance-based classification with lipschitz functions. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Computational Learning Theory and Kernel Machines, 16th Annual Conference on Computational Learning Theory and 7th Kernel Workshop, COLT/Kernel 2003, Washington, DC, USA, August 24-27, 2003, Proceedings*, volume 2777 of *Lecture Notes in Computer Science*, pages 314–328. Springer, 2003.
- [42] Gamaleldin F. Elsayed, Dilip Krishnan, Hossein Mobahi, Kevin Regan, and Samy Bengio. Large margin deep networks for classification. In Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pages 850–860, 2018.
- [43] Kilian Q. Weinberger, John Blitzer, and Lawrence K. Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in Neural Information Processing Systems 18 [Neural Information Processing Systems, NIPS 2005, December 5-8, 2005, Vancouver, British Columbia, Canada]*, pages 1473–1480, 2005.
- [44] Robi Bhattacharjee and Kamalika Chaudhuri. When are non-parametric methods robust? In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 832–841. PMLR, 2020.
- [45] Sadia Chowdhury and Ruth Urner. On the (un-)avoidability of adversarial examples. *CoRR*, abs/2106.13326, 2021. URL <https://arxiv.org/abs/2106.13326>.
- [46] Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher complexity for adversarially robust generalization. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pages 7085–7094. PMLR, 2019. URL <http://proceedings.mlr.press/v97/yin19b.html>.
- [47] Chen Dan, Yuting Wei, and Pradeep Ravikumar. Sharp statistical guarantees for adversarially robust gaussian classification. *CoRR*, abs/2006.16384, 2020. URL <https://arxiv.org/abs/2006.16384>.
- [48] Daniel Cullina, Arjun Nitin Bhagoji, and Prateek Mittal. Pac-learning in the presence of adversaries. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 230–241. Curran Associates, Inc., 2018.
- [49] Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John C. Duchi, and Percy Liang.

- Understanding and mitigating the tradeoff between robustness and accuracy. *CoRR*, abs/2002.10716, 2020. URL <https://arxiv.org/abs/2002.10716>.
- [50] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.
 - [51] Justin Khim and Po-Ling Loh. Adversarial risk bounds for binary classification via function transformation. *CoRR*, abs/1810.09519, 2018. URL <http://arxiv.org/abs/1810.09519>.
 - [52] Idan Attias, Aryeh Kontorovich, and Yishay Mansour. Improved generalization bounds for robust learning. In Aurélien Garivier and Satyen Kale, editors, *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, volume 98 of *Proceedings of Machine Learning Research*, pages 162–183. PMLR, 2019.
 - [53] Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Black-box certification and learning under adversarial perturbations. *CoRR*, abs/2006.16520, 2020. URL <https://arxiv.org/abs/2006.16520>.
 - [54] Ilias Diakonikolas, Daniel M. Kane, and Pasin Manurangsi. The complexity of adversarially robust proper learning of halfspaces with agnostic noise. *CoRR*, abs/2007.15220, 2020. URL <https://arxiv.org/abs/2007.15220>.
 - [55] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Lower bounds on adversarial robustness from optimal transport. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d’Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pages 7496–7508, 2019.
 - [56] Edgar Dobriban, Hamed Hassani, David Hong, and Alexander Robey. Provable tradeoffs in adversarially robust classification. *CoRR*, abs/2006.05161, 2020. URL <https://arxiv.org/abs/2006.05161>.