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

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

LIST OF FIGURES

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 [4, 5, 6]. 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 [7, 8, 9] suggests that this might be true for certain classifiers or algorithms. Finally, others [10, 11, 8] 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 [12], 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 [8, 7].

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 [12, 13], 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 constrast, 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, [14] 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 converges 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 [14] 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 [14] 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 [15, 16, 17, 18, 4] and defenses [19, 20, 2, 21, 22, 23] 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 [24, 25, 8, 14] as well as defenses. Wang et. al. [8] 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. [14] 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 [19] 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 [26, 27, 28]. 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 [29, 30, 31, 32, 33, 34], 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 [35, 13], 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 $\mathscr{X}$ 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 1.2.1. *(Accuracy) Let $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$, and let $f \in \{\pm 1\}^{\mathscr{X}}$ be a classifier. Then the* ***accuracy*** *of f over $\mathscr{D}$, denoted $A(f, \mathscr{D})$, is the fraction of examples $(x, y) \sim \mathscr{D}$ for which $f(x) = y$. Thus*

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

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 1.2.2. *(Robustness) A classifier $f \in \{\pm 1\}^{\mathscr{X}}$ 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* [8, 7].

Definition 1.2.3. *(Astuteness) A classifier $f \in \{\pm 1\}^{\mathscr{X}}$ 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 $\mathscr{D}$, denoted $A_r(f, \mathscr{D})$, is the fraction of examples $(x, y) \sim \mathscr{D}$ for which f is astute at (x, y) with radius r. Thus*

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

It is worth noting that $A_0(f, \mathscr{D}) = A(f, \mathscr{D})$, since astuteness with radius 0 is simply the accuracy. For this reason, we will use $A_0(f, \mathscr{D})$ 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 1.2.4. *(Bayes Optimal Classifier) The* ***Bayes Optimal Classifier*** *on a distribution $\mathscr{D}$, denoted by g^*, is defined as follows. Let $\eta(x) = p_{\mathscr{D}}(+1|x)$. 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 $\mathscr{D}$ over all classifiers.

Definition 1.2.5. *(Consistency) Let M be a classification algorithm over $\mathscr{X} \times \{\pm 1\}$. M is said to be* ***consistent*** *if for any $\mathscr{D}$ over $\mathscr{X} \times \{\pm 1\}$, and any ε, δ over $(0,1)$, there exists N such that for $n \geq N$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$, we have:*

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

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

How can we incorporate robustness in addition to accuracy in this notion? A plausible way, as used in [8], 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 $\mathscr{D}$ over $\mathscr{X} = [0,1]$ such that $\mathscr{D}_{\mathscr{X}}$ 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 $P_{(x,y)\sim\mathscr{D}}[y=+1]=\frac{1}{2}$, it follows that $A_r(f,\mathscr{D})=\frac{1}{2}$.

This motivates the notion of the r-optimal classifier, introduced by [14], which is the classifier with maximum astuteness.

Definition 1.2.6. *(r-optimal classifier) The* ***r-optimal classifier*** *of a distribution G denoted by g_r^* is the classifier with maximum astuteness. Thus*

$$g_r^* = \underset{f\in\{\pm1\}^{\mathscr{X}}}{\arg\max}\, A_r(f,\mathscr{D}).$$

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

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

Definition 1.2.7. *(r-consistent) Let M be a classification algorithm over $\mathscr{X}\times\{\pm1\}$. M is said to be* ***r-consistent*** *if for any $\mathscr{D}$, any $\varepsilon,\delta\in(0,1)$, and $0<\gamma<r$, there exists N such that for $n\geq N$, with probability $1-\delta$ over $S\sim\mathscr{D}^n$,*

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

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

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

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

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

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 1.2.9. *[12] A* ***weight function*** *W is a non-parametric classifier with the following properties.*

1. *Given input $S = \{(x_1, y_1), (x_2, y_2,), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$, W constructs functions $w_1^S, \ldots, w_n^S : \mathscr{X} \to [0,1]$ such that for all $x \in \mathscr{X}$, $\sum_1^n w_i^S(x) = 1$. The functions w_i^S are allowed to depend on $x_1, x_2, \ldots x_n$ but must be independent of $y_1, y_2, \ldots, y_n$.*
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 1.2.10. *A **histogram classifier**, H, is a non-parametric classification algorithm over $\mathbb{R}^d \times \{\pm 1\}$ that works as follows. For a distribution $\mathscr{D}$ over $\mathbb{R} \times \{\pm 1\}$, H takes $S = \{(x_i, y_i) : 1 \leq i \leq n\} \sim \mathscr{D}^n$ as input. Let k_i be a sequence with $\lim_{i\to\infty} k_i = \infty$ and $\lim_{i\to\infty} \frac{k_i}{i} = 0$. H constructs a set of hypercubes $C = \{c_1, c_2, \ldots, c_m\}$ 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 1.2.11. *A **kernel classifier** is a weight function W over $\mathscr{X} \times \{\pm 1\}$ constructed from function $K : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+$ and some sequence $\{h_n\} \subset \mathbb{R}^+$ in the following manner. Given $S = \{(x_i, y_i)\} \sim \mathscr{D}^n$, 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.

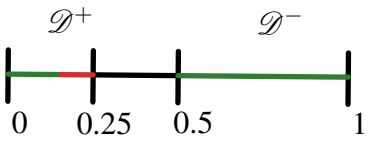


Figure 1.1. H_S is astute in the green region, but not robust in the red region.

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

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

1. $\mathscr{D}$ is 0.1-separated, since the supports of $\mathscr{D}^+$ and $\mathscr{D}^-$ 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 $A_{0.1}(H_S, \mathscr{D}) = 0.8$.

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 1.2.9) to be r-consistent over an r-separated distribution.

Theorem 1.3.1. *Let $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$, and let W be a weight function. Let X be a random variable with distribution $\mathscr{D}_{\mathscr{X}}$, and $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$. Suppose that for any $0 < a < b$,*

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

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

First, we compare Theorem 1.3.1 to Stone's theorem [13], 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 1.3.2. *[13] Let W be weight function over $\mathscr{X} \times \{\pm 1\}$. Suppose the following conditions hold for any distribution $\mathscr{D}$ over $\mathscr{X} \times \{\pm 1\}$. Let X be a random variable with distribution $\mathscr{D}_{\mathscr{X}}$, and $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$. All expectations are taken over X and S.*

1. *There is a constant c such that, for every nonnegative measurable function f satisfying $\mathbb{E}[f(X)] < \infty$,*
$$\mathbb{E}[\sum_{1}^{n} w_i^S(X) f(x_i)] \leq c\mathbb{E}[f(x)].$$
2. *For all $a > 0$,*
$$\lim_{n\to\infty} \mathbb{E}[\sum_{1}^{n} w_i^S(x) I_{||x_i - X|| > a}] = 0,$$
where $I_{||x_i - X|| > a}$ is an indicator variable.
3. $$\lim_{n\to\infty} \mathbb{E}[\max_{1 \leq i \leq n} w_i^S(X)] = 0.$$

Then W is consistent.

There are two main differences between Theorem 1.3.1 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 [13] 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 $X \sim \mathscr{D}$, we require the same to *uniformly* hold over a ball centered at X.

Theorem 1.3.1 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 1.3.1.

Corollary 1.3.3. *Let $\mathscr{D}$ be any r-separated distribution. Let k_n be any sequence such that $\lim_{n\to\infty} \frac{k_n}{n} = 0$, and let M be the k_n-nearest neighbors classifier on a sample $S \sim \mathscr{D}^n$. Then M is r-consistent with respect to $\mathscr{D}$.*

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 $k_n \to \infty$ in addition to $k_n/n \to 0$. 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 1.3.4. *Let W be a kernel classifier over $\mathscr{X} \times \{\pm 1\}$ constructed from K and h_n. Suppose the following properties hold for K and h_n.*

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

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

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

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

W can be shown to satisfy all the conditions of Theorem 1.3.2 (the proof is analogous to the case for a Gaussian Classifier), and is therefore consistent. However, W does not learn a robust classifier on $\mathscr{D}$ for $r = 0.3$.

Consider $x = -0.7$. For any $\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\} \sim \mathscr{D}^n$, 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 $1 \le i \le n$. 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

[14] 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.

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

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

Definition 1.4.1. *[14] Fix $r,\mathscr{D}$. 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 $d(S^*_j,x)\leq r$.*

[14] show that the r-optimal classifier achieves the optimal astuteness – out of all classifiers on the data distribution $\mathscr{D}$; 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, [14] 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 1.4.2. *A set $S_r \subset \mathscr{X} \times \{\pm 1\}$ 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 $AdvPrun(S, r)$ 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 [14] 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:** $S \sim \mathscr{D}^n$, weight function W, robustness radius r;

2 $S_r \leftarrow AdvPrun(S, r)$;

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 $RobustNonPar(W, r)$ represent the weight function that outputs weights for $S \sim \mathscr{D}^n$ according to $RobustNonPar(S, W, r)$. 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 $RobustNonPar(W, r)$ r-consistent? Our next theorem provides sufficient conditions for this.

Theorem 1.4.3. *Let W be a weight function over $\mathscr{X} \times \{\pm 1\}$, and let $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$. Fix $r > 0$. Let $S_r = AdvPrun(S, r)$. For convenience, relabel x_i, y_i so that*

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

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

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

Remark:

There are two important differences between the conditions in Theorem 1.4.3 and Theorem 1.3.1.

1. We replace S with S_r.

2. The expectation over $X \sim \mathscr{D}_{\mathscr{X}}$ is replaced with an average over $\{x_1, x_2, \ldots, x_m\}$. The intuition here is that we are replacing $\mathscr{D}$ with a uniform distribution over S_r. While $\mathscr{D}$ 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 1.4.3. We next show that k_n-nearest neighbors and kernel classifiers with rapidly decaying kernel functions continue to satisfy the conditions in Theorem 1.4.3; this means that these classifiers, when combined with Adversarial Pruning, will converge to r-optimal classifiers in the large sample limit.

Corollary 1.4.4. *Let* k_n *be a sequence with* $\lim_{n\to\infty} \frac{k_n}{n} = 0$*, and let M denote the* k_n*-nearest neighbor algorithm. Then for any* $r > 0$*, RobustNonPar*(M, r) *is r-consistent.*

Remark:

Corollary 1.4.4 gives a formal guarantee in the large sample limit for the modified nearest-neighbor algorithm proposed by [14].

Corollary 1.4.5. *Let W be a kernel classifier over* $\mathscr{X} \times \{\pm 1\}$ *constructed from K and* h_n*. Suppose the following properties hold for K and* h_n*.*

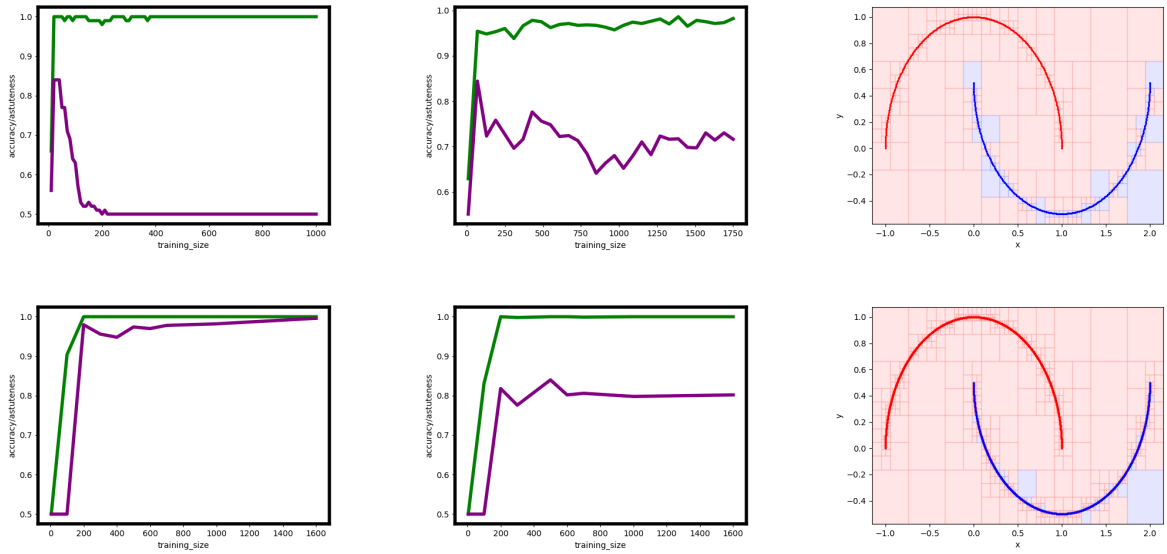


Figure 1.2. Empirical accuracy/astuteness of different classifiers as a function of training sample size. Accuracy is shown in green, astuteness in purple. Left : Noiseless Setting. Right: Noisy Setting. Top Row: Histogram Classifier, Bottom Row: 1-Nearest Neighbor

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

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

Then for any $r > 0$, $RobustNonPar(W, r)$ *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. [14]. For the histogram classifier, we use the optimal attack framework proposed by [14], 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 1.3.1 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 [14] 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 [14] does lead to optimally astute solutions in the large sample limit.

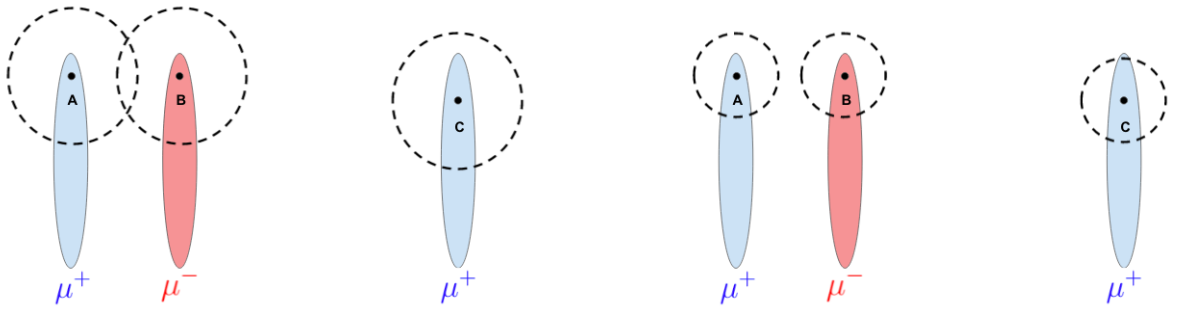
Chapter 2

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 [15, 16, 17, 18, 4, 19, 2, 21, 22, 23, 36], 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



(a) Large robustness radii

(b) Small robustness radii

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 [14] 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 [13] 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 $\mathscr{D} = (\mu, \eta)$ over $\mathbb{R}^d \times \{\pm 1\}$.

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

Our goal is to build classifiers $\mathbb{R}^d \to \{\pm 1\}$ 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 $\mathscr{U} = \{U_x : x \in \mathbb{R}^d\}$ denote the collections of all robustness regions.

We say that a classifier $f : \mathbb{R}^d \to \{\pm 1\}$ 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 2.2.1. *A classifier $f : \mathbb{R}^d \to \{\pm 1\}$ is said to be **astute** at (x,y) with respect to robustness collection $\mathscr{U}$ if $f(x) = y$ and f is robust at x with respect to $\mathscr{U}$. If $\mathscr{D}$ is a data distribution over $\mathbb{R}^d \times \{\pm 1\}$, the **astuteness** of f over $\mathscr{D}$ with respect to $\mathscr{U}$, denoted $A_{\mathscr{U}}(f,\mathscr{D})$, is the fraction of examples $(x,y) \sim \mathscr{D}$ for which f is astute at (x,y) with respect to $\mathscr{U}$. Thus*

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

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, $S = \{(x_1,y_1),(x_2,y_2,),\ldots,(x_n,y_n)\}$ by assigning a function $w_i^S : \mathbb{R}^d \to [0,1]$ 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 $x_1,\ldots,x_n$ but must be independent of the labels $y_1,\ldots,y_n$. 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 $K : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ 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, $\mathscr{U} = \{U_x\}_{x\in\mathbb{R}^d}$, 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.

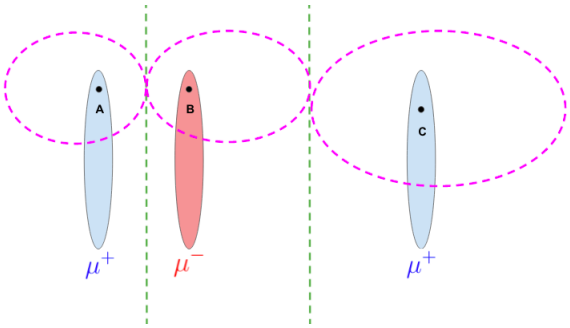


Figure 2.2. The decision boundary of the neighborhood preserving Bayes optimal classifier is shown in green, and the neighborhood preserving robust region of x is shown in pink. The former consists of points equidistant from μ^+, μ^-, and the latter consists of points equidistant from x, μ^+.

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 the neighborhood preserving Bayes optimal classifier, defined as follows.

Definition 2.3.1. *Let $\mathscr{D} = (\mu, \eta)$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then the **neighborhood preserving Bayes optimal classifier of** $\mathscr{D}$, 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 $\mathscr{U} = \{U_x\}$. While these regions U_x can be nearly

arbitrary, we seek regions U_x such that $A_{\mathscr{U}}(g_{max}, \mathscr{D}) = A(g_{bayes}, \mathscr{D})$ (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 2.3.2. *Let $\mathscr{D} = (\mu, \eta)$ 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*

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

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 $\mathscr{D}$ 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 $x \in supp(\mathscr{D})$ 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 2.3.3. *(Accuracy) Let $\mathscr{D}$ be a data distribution. Let $\mathscr{V}$ 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 $A_{\mathscr{V}}(g_{neighbor}, \mathscr{D}) = A(g, \mathscr{D})$, where $A(g, \mathscr{D})$ denotes the accuracy of the Bayes optimal. Thus, $g_{neighbor}$ maximizes accuracy.*

Theorem 2.3.4. *(Robustness) Let $\mathscr{D}$ be a data distribution, let f be a classifier, and let $\mathscr{U}$ be a set of robustness regions. Suppose that $A_{\mathscr{U}}(f,\mathscr{D}) = A(g,\mathscr{D})$, where g denotes the Bayes optimal classifier. Then there exists $x \in supp(\mathscr{D})$ 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 2.3.3 shows that the neighborhood preserving Bayes classifier achieves maximal accuracy, while Theorem 2.3.4 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 2.3.5. *Let $0 < \kappa < 1$ be a real number, and let $\mathscr{D} = (\mu,\eta)$ 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*

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

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 2.3.6. *A learning algorithm A is said to be* ***neighborhood consistent*** *if the following holds for any data distribution* $\mathscr{D} = (\mu, \eta)$ *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* $S \sim \mathscr{D}^n$*,*

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

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 $\mathscr{V}^\kappa$. Observe that as $\kappa \to 1$, 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:

$\lim_{n\to\infty} k_n = \infty$, and $\lim_{n\to\infty} \frac{k_n}{n} = 0$. 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 2.4.1. *Let $\mathscr{D} = (\mu, \eta)$ 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 $\mathscr{D}$.*

The issue in the example above is that for smaller k_n, k_n-nearest neighbors lacks sufficient precision. For neighborhood consistnecy, 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^κ. 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 2.4.2. *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 2.4.3. *Let W be a weight function and let $S = \{x_1, x_2, \ldots, x_n\}$ 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 2.4.4. *Let W be a weight function, $\mathscr{D}$ a distribution over $\mathbb{R}^d \times \{\pm 1\}$, $\mathscr{U}$ 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 $\mathscr{D}$.*
2. *For any $0 < p < 1$, $\lim_{n\to\infty} E_{S\sim\mathscr{D}^n}[\sup_{x\in\mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)}] = 0$.*
3. *$\lim_{n\to\infty} E_{S\sim D^n}[t_n \sup_{x\in\mathbb{R}^d} w_i^S(x)] = 0$.*
4. *$\lim_{n\to\infty} E_{S\sim D^n} \frac{\log T(W,S)}{t_n} = 0$.*

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

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 $x \sim \mathscr{D}$. 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 [37], 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 2.4.4 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 2.4.5. *Suppose* $(k_n)_1^\infty$ *satisfies (1)* $\lim_{n\to\infty} \frac{k_n}{n} = 0$*, and (2)* $\lim_{n\to\infty} \frac{\log n}{k_n} = 0$*. Then* k_n*-nearest neighbors is neighborhood consistent.*

As a result of Theorem 2.4.1, corollary 2.4.5 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 2.4.6. *Let* W *be a kernel classifier over* $\mathbb{R}^d \times \{\pm 1\}$ *constructed from* $K : \mathbb{R}^+ \to \mathbb{R}^+$ *and* h_n*. Suppose the following properties hold.*

1. K *is decreasing, and satisfies* $\int_{\mathbb{R}^d} K(||x||)dx < \infty$.

2. $\lim_{n\to\infty} h_n = 0$ *and* $\lim_{n\to\infty} nh_n^d = \infty$.

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

4. For any $x \geq 0$, $\lim_{n\to\infty} \frac{n}{\log n} K(\frac{x}{h_n}) = \infty$.

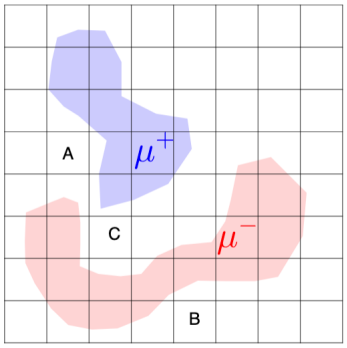


Figure 2.3. we have a histogram classifier being applied to the blue and red regions. The classifier will be unable to construct good labels in the cells labeled A, B, C, and consequently will not be robust with respect to V_x^κ for sufficiently large κ.

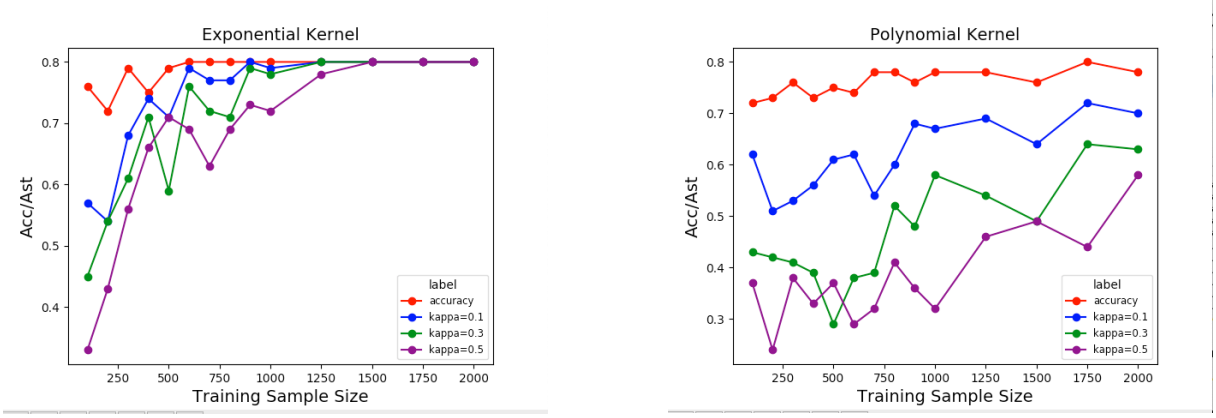
Then W is neighborhood consistent.

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 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^κ will intersect A for some point



(a) exponential kernel

(b) polynomial kernel

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 givne 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 2.4.6. In panel (b), although the accuracy appears to converge, none of the robustness measure. In fact, they get progressively worse the larger κ gets.

$x \in \mu^+$. 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.

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-parametic 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 2.4.6, 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 overlayed circles (see Figure B.1 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 B.4.

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 2.3.5). 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 $n \to \infty$, $A_{\mathscr{A}^\kappa}$ 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 2.4.6, and in particular fails criteria 3. However, note that the polynomial kernel nevertheless performs will 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 2.4.6 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. [15, 16, 17, 18, 4, 19, 20, 2, 21, 22, 23], 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.

[38] and [39] 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 $\mathscr{U}$ induce a *margin constraint* on the decision boundary. The most popular large margin classifier is the Support Vector Machine[40, 41, 42] – 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, [43] 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 [44] for deep neural networks that minimizes the worst case margin, and [45] 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. [24, 25, 8, 14] provide adversarial attacks on non-parametric methods. Wang et. al. [8] develops a defense for 1-NN that removes a subset of the training set to ensure higher robustness. Yang et. al [14] proposes the r-optimal classifier – which is the maximally astute classifier in the standard robustness framework – and proposes a defense called Adversarial Pruning.

Theoretically, [46] 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, [47] 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 $\mathscr{H}$, a robustness metric d, and a radius r. The learner's goal is to find a classifier $h \in \mathscr{H}$ 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 $\mathscr{H}$ is the class of linear classifiers, and d is an ℓ_p-metric? Prior work has provided both upper [48, 1] as well as lower bounds [2, 1] 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 [36], 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 [49] 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 [48] 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 [15, 16, 17, 18, 4, 19, 20, 21, 22, 23] empirically studying adversarial examples primarily in the context of neural networks. Several works [2, 50, 51] 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 [52, 53, 9, 48, 54] have focused on finding distribution agnostic bounds of the sample complexity for robust classification. In [9], 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.

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

In [48], 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 it's simplicity and elegance, the case where the data distribution is a mixture of Gaussians has been particularly well-studied. The first such work was [2], 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 [56], [57] and [1]. [56] 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. [57] applies expands upon [2] by consider mixtures of three Gaussians in both the ℓ_2 and ℓ_∞ norms. Finally, [1] fully generalizes the results of [2] 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 $p \in [1, \infty]$). [2] and [1] bear the most relevance with our work, and we consequently carefully compare our results in section 3.3.1.

Another approach for lower and upper bounds on sample complexities for linear classifiers can be found in [49], 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, [49] 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. [8] considers the nearest neighbors algorithm, and shows how to adapt it for converging towards a robust classifier. In [14], 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. [46] 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 $diam_p(S)$ denote its diameter: that is, $diam_p(S) = \sup_{x,y \in S} ||x-y||_p$.

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 3.2.1. *Let $\mathscr{D}$ 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 $\mathscr{D}$, denoted $\mathscr{L}(f, \mathscr{D})$, is the fraction of examples $(x,y) \sim \mathscr{D}$ for which f is not accurate. Thus*

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

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

Definition 3.2.2. *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 3.2.3. *The **robust loss** of f over $\mathscr{D}$, denoted $\mathscr{L}_r(f, \mathscr{D})$, is the fraction of examples $(x,y) \sim \mathscr{D}$ 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*

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

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

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

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

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

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

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 3.2.5. *Let $\mathscr{D}$ be a distribution over $\mathbb{R}^d\times\{\pm1\}$ and A be a learning algorithm. Then the* ***standard sample complexity*** *of A with respect to $\mathscr{D}$, denoted $m^\varepsilon(A,\mathscr{D})$, is the minimum number of training samples needed such that A has expected standard loss at most ε. Formally,*

$$m^\varepsilon(A,\mathscr{D}) = \min(\{n : E\mathscr{L}^n(A,D)\leq\varepsilon\}).$$

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

$$m_r^\varepsilon(A,\mathscr{D}) = \min(\{n : E\mathscr{L}^n(A,D)\leq\varepsilon\}).$$

3.2.3 Linear classifiers

In this work, we consider linear classifiers, formally defined as follows:

Definition 3.2.6. *Let $w \in \mathbb{R}^d$ be a vector. Then the **linear classifier** with parameters $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$ over $\mathbb{R}^d \times \pm 1$, denoted $f_{w,b}$, is defined as ,*

$$f_{w,b}(x) = \begin{cases} +1 & \langle w, x \rangle \geq b \\ -1 & \langle w, x \rangle < b \end{cases}.$$

Learning linear classifiers is well understood in the standard classification setting. We now consider the linearly *separable* case, in which some linear classifier has perfect accuracy. We will later define linear r-separability as the robust analog of separability.

Definition 3.2.7. *A distribution $\mathscr{D}$ over $\mathbb{R}^d \times Y$ is **linearly separable** if its support can be partitioned into sets S^+ and S^- such that:*

1. S^+ and S^- correspond to the positively and negatively labeled subsets of $\mathbb{R}^d$. In particular, $P_{(x,y)\sim\mathscr{D}}[x \in S^y] = 1$.

2. There exists a linear classifier, $f_{w,b}$, that has perfect accuracy. That is, $\mathscr{L}(f_{w,b}, \mathscr{D}) = 0$.

The standard sample complexity for linearly separable distributions can be characterized through their margin, which is defined as follows.

Definition 3.2.8. *Let $\mathscr{D}$ be a linearly separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let S^+ and S^- be as above. Then $\mathscr{D}$ has **margin** γ if γ is the largest real number such that there exists a linear classifier $f_{w,b}$ with the following properties:*

1. $f_{w,b}$ has perfect accuracy. That is, $\mathscr{L}(f_{w,b}, \mathscr{D}) = 0$.

2. Let $H_{w,b} = \{x : \langle x, w \rangle = b\}$ denote the decision boundary of $f_{w,b}$. Then for all $x \in (S^+ \cup S^-)$, x has ℓ_2 distance at least γ from $H_{w,b}$. That is,

$$\inf_{x \in S^+ \cup S^-, z \in H_{w,b}} ||x - z||_2 \geq \gamma.$$

We let $\gamma(\mathscr{D})$ denote the margin of $\mathscr{D}$.

Observe that although we use a general norm, ℓ_p, to measure robustness, the margin is always measured in ℓ_2. This is because the ℓ_2 norm plays a fundamental role in bounding the number of samples needed to learn a linear classifier.

The basic idea is that when the ℓ_2 margin is large relative to the ℓ_2 diameter of the distribution, the max margin algorithm requires fewer samples needed to learn a linear classifier. In particular, the ratio between the ℓ_2 margin and the ℓ_2 diameter fully characterizes the standard sample complexity of the max margin algorithm. To further simplify our notation, we define this ratio as the aspect ratio.

Definition 3.2.9. *Let $\mathscr{D}$ be a linearly separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then the **aspect ratio** of $\mathscr{D}$, $\rho(\mathscr{D})$ is defined as,*

$$\rho(\mathscr{D}) = \frac{diam_2(S^+ \cup S^-)}{\gamma(\mathscr{D})},$$

where $diam_2(S^+ \cup S^-)$ denotes its diameter in the ℓ_2 norm.

We now have the following well-known result, which characterizes the expected standard loss with the aspect ratio.

Theorem 3.2.10. (Chapter 10 in [58]) *Let M denote the hard margin SVM algorithm. If $\mathscr{D}$ is a distribution with aspect ratio $\rho = \rho(\mathscr{D})$, then for any $n > 0$ we have $\mathbb{E}_{S \sim \mathscr{D}^n} \mathscr{L}(M_S, \mathscr{D}) \leq O(\frac{\rho^2}{n})$, where M_S denotes the classifier learned by M from training data S.*

We can also express this result in terms of standard sample complexity.

Corollary 3.2.11. *Let M denote the hard margin SVM algorithm. If $\mathscr{D}$ is a distribution with aspect ratio $\rho = \rho(\mathscr{D})$, then for any $\varepsilon > 0$ we have $m^{\varepsilon}(M_S, \mathscr{D}) \leq O(\frac{\rho^2}{\varepsilon})$, where M_S denotes the classifier learned by M from training data S.*

Theorem 3.2.10 and Corollary 3.2.11 will serve as a benchmark for comparison with the robust sample complexity.

3.2.4 Linear r-separability

Finally, we introduce linear r-separability, which is the key characteristic of distributions considered in this paper. This can be thought of as the robust analog of linear separability.

Definition 3.2.12. *For any $r > 0$, a distribution $\mathscr{D}$ over $\mathbb{R}^d \times \{\pm 1\}$ is **linearly r-separable** if there exists a linear classifier $f_{w,b}$ such that $\mathscr{L}_r(f_{w,b}, \mathscr{D}) = 0$.*

This definition is the fundamental property considered in this paper. Our goal is to understand the sample complexity required for learning robust linear classifiers on linearly r-separable distributions, and compare it with the standard sample complexity given in Theorem 3.2.10.

3.3 Lower Bounds

In this section, we consider r-separated distributions whose aspect ratio is constant. By Theorem 3.2.10, the standard sample complexity for learning them is independent of d. We will show that in contrast, the robust sample complexity has a linear dependence on d, and consequently establish a substantial gap between the standard and robust cases.

We begin by defining the family of such distributions.

Definition 3.3.1. *For any ρ, r, the set $\mathscr{F}_{r,\rho}$ is defined as the set of all distributions $\mathscr{D}$ over $\mathbb{R}^d \times \{\pm 1\}$ such that $\mathscr{D}$ is r-separated and has aspect ratio at most ρ.*

We now state our main result.

Theorem 3.3.2. *Let $r > 0$ and $\rho > 20$. Then the following hold.*

1. *For every learning algorithm A, and any $n > 0$, there exists $\mathscr{D} \in \mathscr{F}_{r,\rho}$ such that the expected robust loss when A is trained on a sample of size n from $\mathscr{D}$ is at least $\Omega(\frac{d}{n})$. Formally, there exists a constant $c > 0$ such that $\mathbb{E}_{S \sim \mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})] \geq \frac{cd}{n}$.*

2. *In contrast, by Theorem 3.2.10, for any $\mathscr{D} \in \mathscr{F}_{r,D}$, the max margin algorithm has expected standard loss $O(\frac{\rho^2}{n})$, when trained on a sample of size n from $\mathscr{D}$. Formally, there exists a constant $c' > 0$ such that $\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}(A_S, \mathscr{D})] \leq \frac{c'\rho^2}{n}$.*

The condition $\rho > 20$ is required to rule out degenerate cases. This is because for small values of ρ, the ℓ_2 diameter of $\mathscr{D}$ is not much larger than the ℓ_2 margin of $\mathscr{D}$. This forces $\mathscr{D}$ to be mostly clustered around a line which leads to more complicated behavior.

Observe that when ρ is a constant independent of d, the expected standard loss is $O(\frac{1}{n})$ while the expected robust loss is $\Omega(\frac{d}{n})$. Thus, the ratio between the expected robust loss and the expected standard loss is $\Omega(d)$, leading to a dimensional dependent gap between the robust and standard cases.

We also note that these bounds hold regardless of which ℓ_p ($p \in (1,\infty]$) norm is being used. This is because our construction of $\mathscr{D} \in \mathscr{F}_{r,\rho}$ for which the lower bound holds is given in terms of the norm p. More generally, the family $\mathscr{F}_{r,\rho}$ is implicitly defined with respect to p.

Furthermore, our lower bound differs from the lower bound of $\Omega(\frac{d}{n})$ shown in prior work [49] because it specifically holds for $\mathscr{F}_{r,\rho}$, a linearly r-separated family of distributions with constant aspect ratio. Thus, while [49] has shown the existence of distributions satisfying the first condition of Theorem 3.3.2, our result is the first to exhibit a distribution satisfying both conditions.

Finally, we note that Theorem 3.3.2 can also be expressed in terms of sample complexities. We include this in the following corollary.

Corollary 3.3.3. *Let $r > 0$ and $\rho > 20$. Then the following hold.*

1. For every learning algorithm A, and any $\varepsilon > 0$, there exists $\mathscr{D} \in \mathscr{F}_{r,\rho}$ such that the robust sample complexity of A with respect to $\mathscr{D}$ is at least $\Omega(\frac{d}{\varepsilon})$. Formally, there exists a constant $c > 0$ such that $m_r^{\varepsilon}(A, \mathscr{D}) \geq \frac{cd}{\varepsilon}$.

2. In contrast, by Theorem 3.2.10, for any $\mathscr{D} \in \mathscr{F}_{r,D}$, the max margin algorithm has standard sample complexity $O(\frac{\rho^2}{\varepsilon})$. Formally, there exists a constant $c' > 0$ such that $m^{\varepsilon}(A, \mathscr{D}) \leq$

$\frac{c'\rho^2}{\varepsilon}$.

3.3.1 Comparison with [1] and [2]

The first work to provide a robust sample complexity lower bound that applied to linear classifiers is [2]; they showed a gap of $\Omega(\sqrt{d})$ between the robust and accuracy loss for a specific mixture of two Gaussians. This was later generalized to mixtures of any two Gaussians by [1], who also established more general lower bounds for any ℓ_p norm. Since [1] is a strict generalization of [2], we next explain how our lower bounds differ from [1], and why their techniques do not lead to our results. We begin by summarizing their results.

Summary of [1]

[1] considers data distributions $\mathscr{D}$ that are parametrized by $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$, $\Sigma \succcurlyeq 0$. $\mathscr{D}_{\mu,\Sigma}$ is the mixture of two Gaussians, $\mathscr{N}(\mu,\Sigma)$ and $\mathscr{N}(-\mu,\Sigma)$, with equal mass, where instances drawn from $\mathscr{N}(\mu,\Sigma)$ are labeled as $+$, and instances drawn from $\mathscr{N}(-\mu,\Sigma)$ are labeled as $-$. They consider robustness measured in any normed metric in $\mathbb{R}^d$, including the ℓ_p norm for $p \in (1,\infty]$. Although their bounds apply to any classifier, this effectively deals with linear classifiers since it can be shown that the optimally robust and accurate classifiers are both linear.

For any distribution $\mathscr{D}_{\mu,\Sigma}$, let L_{rob} denote the optimal robust loss of any classifier on $\mathscr{D}_{\mu,\Sigma}$, and let L_{std} denote the optimal standard loss. Then the bounds shown in [1] can restated as follows (a detailed derivation from [1] appears in Appendix C.1).

Theorem 3.3.4. *[1]*

1. *For any learning algorithm A and any $n > 0$, there exists some mixture of Gaussians, $\mathscr{D}_{\mu,\Sigma}$ such that the expected excess robust loss is at least $\Omega(L_{rob}\frac{d}{n})$, when A is trained on a sample of size n from $\mathscr{D}$.*

2. *For any distribution $\mathscr{D}_{\mu,\Sigma}$, it is possible to learn a classifier with expected excess standard loss at most $O(L_{std}\frac{d}{n})$.*

3. *By (1.) and (2.), the ratio between the expected excess loss and expected excess standard loss can be expressed as* $ratio \geq \Omega(\frac{L_{rob}}{L_{std}})$.

Observe that their bounds are given through *excess* losses, which is the amount by which the loss exceeds to the optimal loss. This is necessary because in their setting, the optimal classifiers do not have 0 loss.

Comparison with our bounds

Recall that in our work, we are concerned with the *linearly r-separated case*, which occurs precisely when the optimal robust and standard losses both equal 0. However, from Theorem 3.3.4, we see that although [1] proves a gap between standard and robust sample complexity, this gap is predicated on distributions for which the optimal robust loss, L_{rob} and optimal standard loss, L_{std} differ. Furthermore, in the case where they obtain a gap of $\Omega(d)$, we see that this requires $\frac{L_{rob}}{L_{std}} = \Omega(d)$ which is a substantial difference. By contrast, our results characterize a gap exclusively in the case that this does not occur.

Finally, in the limiting case where the Gaussians they consider are sufficiently far apart, their data will begin to appear linearly r-separated, meaning both L_{rob} and L_{std} are close to 0. However, even in this case, it can be shown that the ratio $\frac{L_{rob}}{L_{std}}$ diverges towards infinity, meaning that their lower bound characterizes a very different dynamic from ours. Precise details on this comparison can be found in appendix C.1.

3.3.2 Intuition behind Theorem 3.3.2

The proof idea for Theorem 3.3.2 can be summarized with a simple example (Figure 3.1). In this example, we seek to learn a linear classifier for a linearly r-separated distribution in $\mathbb{R}^2$. The key idea is to contrast the necessary conditions for learning a robust classifier, and the necessary conditions for learning an accurate classifier.

Observe that the distribution is *precisely* linearly r-separated, that is, it is not possible to achieve robustness for radii larger than r. Because of this, there is a unique linear classifier f_{rob}

that has perfect robustness. In order to learn this classifier, we must see examples from $S^+ \cup S^-$ that are close to the "boundary" of $S^+ \cup S^-$. In our figure, this consists of points that are close to the dotted blue and red lines. Moreover, it can be shown that the number of such examples we must see is related to d, the dimension.

By contrast, any classifier that separates S^+ from S^- has perfect accuracy (take for example f_{std} shown in the figure). It is possible to exploit this by using margin based algorithms for learning linear classifiers. In particular, we no longer need to see points that are extremely close to the boundary of $S^+ \cup S^-$.

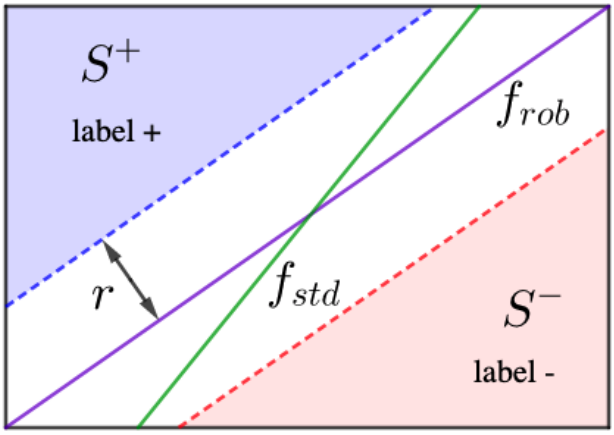


Figure 3.1. An example of a linearly r-separated distribution, with positively and negatively labeled examples in S^+ and S^- respectively. The optimally robust classifier, f_{rob} is shown in purple, while the (not necessarily unique) optimally accurate classifier, f_{std}, is shown in green.

General Hypothesis Classes:

We now briefly consider how to extend our methods to other hypothesis classes. For any hypothesis class $\mathscr{H}$ and distribution $\mathscr{D}$ let

$$\mathscr{H}_{\mathscr{D},\alpha} = \{h : h \in \mathscr{H}, \mathscr{L}(h, \mathscr{D}) \leq \alpha\}$$

and let

$$\mathscr{H}^r_{\mathscr{D},\alpha} = \{h : h \in \mathscr{H}, \mathscr{L}_r(h, \mathscr{D}) \leq \alpha\}.$$

$\mathscr{H}_{\mathscr{D},\alpha}$ can be thought of as the set of accurate classifiers while $\mathscr{H}^r_{\mathscr{D},\alpha}$ can be thought of as the set of astute classifiers. By their definitions, it is clear that $\mathscr{H}^r_{\mathscr{D},\alpha} \subseteq \mathscr{H}_{\mathscr{D},\alpha}$. However, in the case when $\mathscr{H}$ is the set of linear classifiers, we see that for small α, $\mathscr{H}^r_{\mathscr{D},\alpha}$ is a much "smaller" set than $\mathscr{H}_{\mathscr{D},\alpha}$. By exploiting the geometric structure inherent to $\mathscr{H}$, we can much more efficiently search for some $h \in \mathscr{H}_{\mathscr{D},\alpha}$ than we can in $\mathscr{H}^r_{\mathscr{D},\alpha}$. This dynamic is the crux of our lower bound: as we essentially show that there are far more critical points (i.e. points near the decision boundary) that we must see for learning $\mathscr{H}^r_{\mathscr{D},\alpha}$ that aren't required for $\mathscr{H}_{\mathscr{D},\alpha}$.

Thus, for our methods to extend to an arbitrary hypothesis class, we would require a similar dynamic. We need two properties to hold: (1) $\mathscr{H}^r_{\mathscr{D},\alpha}$ must be a very strict subset of $\mathscr{H}_{\mathscr{D},\alpha}$ for sufficiently small alpha. (2) We must have some kind of exploitable geometric structure about $\mathscr{H}$ which allows us to exploit this gap. For the case of linear classifiers, this was the ℓ_2 measured aspect ratio, $\gamma(\mathscr{D})$.

Kernel Classifiers:

A natural choice of a more general hypothesis class would be Kernel Classifiers, which are linear classifiers that operate in an embedded space, H. The main difficulty in expanding our lower bound to this more general setting comes from the behavior near the margin: the effects of the robustness radius in the embedded space are considerably less behaved than they are in the

standard linear case. Nevertheless, we leave this as an important avenue for future work.

Algorithm 2: Adversarial-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$,;

$w \leftarrow 0$;

for $i = 1 \ldots n$ **do**

  $z = \arg\min_{||z - x_i||_p \leq r} y_i \langle w, z \rangle$ finds adversarial example;

  if $\langle w, y_i z \rangle \leq 0$ *checks label* **then**

    $w \leftarrow w + y_i z$ perceptron update;

  end if

end for

Return: $f_{w,0}$;

3.4 Upper Bounds

In the previous section, we showed that for any algorithm, there is some distribution $\mathscr{D} \in \mathscr{F}_{r,\rho}$ that is difficult (i.e. requires high sample complexity) to learn robustly. A natural follow-up question is: what about distributions for which the margin, γ is very large compared to r.

Observe that in Figure 3.1 the robustness radius r is very close to the margin. In particular, we can find adversarial examples from S^+ and S^- that are very close to the decision boundary f_{rob}. By contrast, if $\gamma >> r$, then this no longer holds which suggests that better robust sample complexities might be possible.

In this section, we will describe a subset of $\mathscr{F}_{r,\rho}$ that can be learned with expected loss $O(\frac{1}{n})$, thus matching the standard sample complexity up to a constant factor. To do so, we will introduce a novel concept: the *robust margin*. The basic intuition is that distributions for which the margin greatly exceeds the robustness radius are precisely distributions with a large robust margin. We use the following notation.

Observe that if $\mathscr{D}$ is a linearly r-separated distribution, then $\mathscr{D}$ must also be linearly separable. As earlier, let $S^+, S^- \subset \mathbb{R}^d$ denote the positively and negatively labeled examples from $\mathscr{D}$. We now define

$$S_r^+ = \cup_{s \in S^+} B_p(s,r) \text{ and } S_r^- = \cup_{s \in S^-} B_p(s,r). \tag{3.1}$$

It follows that the decision boundary of any linear classifier with perfect robustness over $\mathscr{D}$ must separate S_r^+ and S_r^-. We now define the robust margin as a measurement of this separation.

Definition 3.4.1. *Let $\mathscr{D}$ be a linearly r-separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let S_r^+ and S_r^- be as above. Then $\mathscr{D}$ has **robust margin** γ_r if γ_r is the largest real number such that there exists a linear classifier $f_{w,b}$ with the following properties:*

1. $f_{w,b}$ has perfect astuteness. That is, $\mathscr{L}_r(f_{w,b}, \mathscr{D}) = 0$.

2. Let $H_{w,b} = \{x : \langle x, w \rangle = b\}$ denote the decision boundary of $f_{w,b}$. Then for all $x \in (S_r^+ \cup S_r^-)$, x has ℓ_2 distance at least γ from $H_{w,b}$. That is,

$$\inf_{x \in S_r^+ \cup S_r^-} \inf_{z \in H_{w,b}} ||x - z||_2 \geq \gamma.$$

We let $\gamma_r(\mathscr{D})$ denote the margin of $\mathscr{D}$, and say that such a distribution is r, γ_r-separated.

It is crucial to note that although adversarial perturbations are measured in ℓ_p, the robust margin is measured in ℓ_2. This is because while the metric ℓ_p plays a role in constructing $B(x,r)$, it can be completely disregarded once the sets S_r^+ and S_r^- are considered, as any hyperplane separating S_r^+ and S_r^- will have perfect robustness.

We now define the robust aspect ratio, which is the robust analog of standard aspect ratio.

Definition 3.4.2. *Let $\mathscr{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then the **robust aspect ratio** of $\mathscr{D}$, $\rho_r(\mathscr{D})$ is defined as*

$$\rho_r(\mathscr{D}) = \frac{diam_2(S_r^+ \cup S_r^-)}{\gamma_r(\mathscr{D})},$$

where as before, $diam_2(S_r^+ \cup S_r^-)$ denotes its diameter in the ℓ_2 norm.

We will now show that just as the aspect ratio, $\rho(\mathscr{D})$, characterized the sample complexity for standard classification, the robust aspect ratio, $\rho_r(\mathscr{D})$ will characterize the sample complexity for robust learning. To do so, we present a perceptron-inspired algorithm (Algorithm 2) for learning a robust classifier on r-separated data with robust aspect ratio ρ_r.

The basic idea behind Algorithm 2 is to combine the standard perceptron algorithm with adversarial training. In particular, we iterate through the training set and do the following on each point (refer to Algorithm 2 for precise details).

1. Find an adversarial example (z, y_i) by attacking our classifier, $f_{w,0}$, at (x_i, y_i) (line 4). This is a straightforward convex optimization problem for linear classifiers.

2. If $f_{w,0}(z) \neq y_i$, we update our weight vector with (z, y_i) by using the standard perceptron update (lines 5-6).

We have the following upper bound on the expected robust loss of our algorithm.

Theorem 3.4.3. *Let $\mathscr{D}$ be a distribution with robust aspect ratio $\rho_r(\mathscr{D})$. Then for any $n > 0$, we have*

$$\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})] \leq O(\frac{\rho_r(\mathscr{D})^2}{n}),$$

where A_S denotes the classifier learned by Algorithm 2 from training data S.

Observe that this expected loss is still larger than the expected standard loss in Theorem 3.2.10 as $\rho_r(\mathscr{D}) > \rho(\mathscr{D})$ for any $\mathscr{D}$. We also note that this result is not contradictory with our lower bound; there exist distributions $\mathscr{D} \in \mathscr{F}_{r,\rho}$ such that $\gamma_r(\mathscr{D}) = 0$, and these are precisely the distributions for which our lower bounds hold.

3.4.1 Generalization to Kernel Classifiers

Algorithm 2 can be thought of as the robust analog to the perceptron algorithm. We now generalize this algorithm to obtain a robust variant of the *kernel perceptron algorithm*. We first

briefly review kernel classifiers. A detailed explanation of our generalized algorithm along with requisite background material can be found in Appendix C.4

Definition 3.4.4. *Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ be a kernel similarity function, $T = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathbb{R}^d \times \{\pm 1\}$ be a set of labeled points, and $\alpha \in \mathbb{R}^m$ be a vector of m real numbers. Then the* ***kernel classifier*** *with similarity function K, parameters T, α, and denoted by $f^{\alpha}_{T,K}$ is defined as*

$$f^{K}_{T,\alpha}(x) = \begin{cases} +1 & \sum_1^m \alpha_i y_i K(x_i, x) \geq 0 \\ -1 & \sum_1^m \alpha_i y_i K(x_i, x) < 0 \end{cases}.$$

Conceptually, kernel classifiers are linear classifiers operating in embedded space. With each kernel similarity function K, there is a map $\phi : \mathbb{R}^d \to H$ (where H is some Hilbert space) such that $K(x, x') = \langle \phi(x), \phi(x') \rangle$. Thus we can think of kernel classifiers as having a linear decision boundary in H.

We now present an analog of Algorithm 2 that we call the Adversarial Kernel-Perceptron. The essence of this algorithm has not changed. For each (x_t, y_t) in our training set, we do the following.

1. Find an adversarial example (z, y_i) by attacking our classifier, $f^{K}_{T,\alpha}$, at (x_i, y_i) (line 4).
2. If $f^{K}_{T,\alpha}(z) \neq y_i$, we update our weight vector with (z, y_i) by appending (z, y_i) to T lines

(5-6). This corresponds to a kernel-perceptron update that uses (z, y_i) instead of (x_i, y_i).

Algorithm 3: Adversarial-Kernel-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$, Similarity function, K
$T \leftarrow \emptyset$, $\alpha \leftarrow 0$
for $i = 1 \ldots n$ **do**
  $z = \arg\min_{||z-x||_p \leq r} y_i f^K_{T,\alpha}(z)$ finds adv. ex.
  if $f^k_{T,\alpha}(z) \leq 0$ *checks label* **then**
    $T = T \cup \{(z, y_i)\}$ kern. percep. update
    $\alpha = (1, \ldots, 1)_{|T|}$
  end if
end for
Return $f^K_{T,\alpha}$

One challenging aspect of this algorithm is minimizing $f^k_{T,\alpha}(z)$. For linear classifiers, this has a closed form solution that utilizes the dual norm. For arbitrary Kernel classifiers, this is a somewhat more challenging problem. However, we note that this can be solved using standard optimization techniques, and in some cases (when K is particularly simple), it can be solved with basic gradient descent.

Finally, we show that this Algorithm has similar performance to the linear case. Instead of using the robust aspect ratio, $\rho_r(\mathscr{D})$, to bound the performance, we will require the **robust K-aspect ratio**, which is the kernel analog of this quantity. It can be thought of as the robust aspect ratio in the embedded space H. Details about this quantity (along with the proof of the theorem) can be found in Appendix C.4.

Theorem 3.4.5. *Let $\mathscr{D}$ be a distribution with robust K-aspect ratio $\rho^K_r(\mathscr{D})$. Then for any $n > 0$, we have*

$$\mathbb{E}_{S \sim \mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})] \leq O(\frac{\rho^K_r(\mathscr{D})^2}{n}),$$

where A_S denotes the classifier learned by Algorithm 3 from training data S.

This result indicates that for small values of $\rho_r^k(\mathscr{D})$, we can achieve a very good robust sample complexity for kernel classifiers. However, as the size of the perturbations approach this margin, this quantity goes to infinity. This phenomenon mirrors the linearly separable case, and suggests that a similar overall dynamic holds for kernel classification. We leave finding a full generalization (including our lower bound) for a direction in future work.

Chapter 4

Robust Empirical Risk Minimization with Tolerance

4.1 Introduction

Adversarially robust classification is a staple of modern machine learning. In the robust setting, along with meeting standard accuracy guarantees, predictions made by a learner at test time must additionally be robust to adversarial perturbations to the input, typically defined by a fixed family $\mathscr{U} = \{U_x\}_{x \in X}$ of possible perturbations. Developing robust algorithms with provable guarantees has been an important research direction in recent years, both for parametric [52, 53, 9, 48, 54] and non-parametric [8, 14, 46, 59] classifiers, but understanding the performance of even the most basic algorithms in the setting remains open.

In this work, we study one of the simplest, most fundamental algorithmic paradigms in learning, a classical method called *empirical risk minimization* (ERM). In the robust setting, an algorithm is said to be an empirical risk minimizer (RERM) if it always outputs a hypothesis in the class with minimal *robust* risk over its training data. In the standard setting, it is a classical result that any learnable class is learnable (near-optimally) by any ERM. Unfortunately, this is known to fail drastically in the robust setting—Montasser et al. [9] showed that there exist finite VC classes, $\mathscr{H}$, where no algorithm outputting hypotheses in $\mathscr{H}$ (called a *proper* learner) can converge towards the optimal classifier, even with arbitrary amounts of training data. Conversely, such classes *are* in fact robustly learnable, but require complicated improper learning rules and a

potentially exponential number of samples.

The failure of Robust ERM for general classes raises an interesting question: *are there natural sufficient conditions for the success of RERM?* One obvious answer to this question is the notion of robust VC dimension, a combinatorial parameter promising the success of RERM. However, bounding robust VC is typically difficult, and such results are only known for very specialized examples of classifiers and robustness regions (e.g. linear classifiers under fixed-radius balls [49] or other simple margin structures [54], or VC-classes under finite perturbation sets [53]). To our knowledge there are no corresponding results for more general robustness regions and hypothesis classes beyond these special cases.

Given the current failure of combinatorial techniques in this setting, one might instead hope to show RERM works given sufficiently nice *geometric* conditions on the hypothesis class. Sadly, this is not the case. We show that there exist robustness regions for which RERM (indeed any proper algorithm) fails even for settings as simple as (bounded) linear classifiers.

Theorem 4.1.1 (Failure of RERM for Linear Classifiers)**.** *For any $W > 0$ and $d > 1$, let $\mathscr{H}_W$ denote the set of linear classifiers with distance at most W from the origin. Then there exists a set of robustness regions U over $\mathbb{R}^d$ such that for any proper learning algorithm L there exists a distribution $\mathscr{D}$ for which the following hold:*

- *$\mathscr{D}$ **is realizable**: There exists $h^* \in \mathscr{H}_W$ such that $\ell_U(h^*, \mathscr{D}) = 0$.*
- *L **has high error**: With probability at least $\frac{1}{7}$ over $S \sim \mathscr{D}^m$, $\ell_U(L(S), \mathscr{D}) > \frac{1}{8}$.*

With this in mind, we turn our attention to a different approach: relaxing the notion of robustness itself. We'll consider a recent model of Ashtiani et al. [60] called *tolerant* robust learning. In the tolerant setting, the learner is only required to compete with the best loss over a relaxed family of perturbation sets $\mathscr{U}^\gamma$ for a (potentially arbitrary) tolerance parameter $\gamma > 0$. Ashtiani et al. [60] studied this setting in the special case of radius r balls, where the learner competes with robust error against $r(1+\gamma)$-balls. Under this framework, Ashtiani et al. [60]

give an algorithm with PAC-guarantees for VC classes using significantly fewer samples, but their techniques remain improper and only hold for the simplest robustness setting.

In this work, we show that a simple variant of RERM in the tolerant model indeed succeeds under natural geometric conditions on the hypothesis class. In particular, we study a notion of smoothness called *regularity*, which roughly promises that every point in the instance space should be contained in some ball of the same label. This captures many well-studied settings, such as cases where the decision boundaries are compact, differential manifolds in $\mathbb{R}^d$.

Theorem 4.1.2 (Tolerant RERM for Regular Classes)**.** *Let $\mathscr{H}$ be a regular hypothesis class with VC dimension v over $\mathbb{R}^d$, and let $\mathscr{U}$ be any set of robustness regions. Then TolRERM tolerantly PAC-learns $(\mathscr{H}, \mathscr{U})$ with tolerant sample complexity*

$$m(\varepsilon, \delta, \gamma) = O\left(\frac{vd \log \frac{dDiam(U)}{\varepsilon\gamma\delta}}{\varepsilon^2}\right),$$

where $Diam(U)$ denotes the maximum ℓ_2 diameter across robustness regions U_x.

Theorem 4.1.2 matches the sample complexity given in Ashtiani et al. [60] up to logarithmic factors and enjoys the additional benefits of applying to more general robustness regions along with its properness and general algorithmic simplicity. For completeness, we also analyze our algorithm's performance over non-regular classifiers in Appendix D.4, and show that it has a similar performance albeit at the cost of replacing the VC-dimension with v_{ball}, the robust VC dimension of $\mathscr{H}$ over balls of a fixed radius. Thus, for non-regular hypothesis classes, our algorithm gives a reduction from arbitrary robustness regions to the case where they are all balls of a fixed radius.

Finally it's worth noting that while Ashtiani et al. [60] only requires sampling access to the perturbation sets, stronger access such as an empirical risk minimizer is inevitable in the general setting where $\mathscr{U}$ is unknown. We show that there exists hypothesis classes where $\Omega((\frac{D}{\gamma})^d)$ queries to a sampling oracle are required for robust learning with tolerance if no other

interaction with U_x is permitted.

While Theorem 4.1.2 gives a natural sufficient condition for the success of RERM in relaxed settings, many questions in this direction remain wide open. It would be interesting to identify a necessary condition for the success of RERM, both in the tolerant and original robust models. Furthermore, it should be noted that while we prove RERM fails to learn nice classes in the latter, the perturbation family we use to achieve this is highly combinatorial. As such, there is still hope that RERM may be sufficient in the traditional setting under *joint* niceness conditions on $\mathscr{H}$ and $\mathscr{U}$, though the close interplay between the two families seems to make identifying such a condition difficult, if it is indeed possible at all.

4.2 Related Work

Much of the work on adversarial robustness [15, 16, 17, 18, 4, 19, 20, 21, 22, 23] is done in the context of neural networks.

On the theoretical side, there has been a recent focus on developing algorithms with guarantees in convergence towards an optimal classifier. On the parametric side, several works [52, 53, 9, 48, 54, 49] have focused on distribution agnostic bounds on the amount of data required to converge towards the optimal classifier in a given hypothesis class. For example, Montasser et al. [9] 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. There has also been some work considering the computation complexity required for robust learning such as Diakonikolas et al. [55].

Aside from Ashtiani et al. [60], there are several works which also consider variations on robust learning with tolerance. Yang et al. [14] and Bhattacharjee and Chaudhuri [46] show that certain non-parametric algorithms exhibit a type of tolerant behavior when robustness regions are constrained to be balls of radius r. Montasser et al. [61] considers robustness in the *transductive learning setting*. Their work employs a similar idea to Ashtiani et al. [60] in that they consider

expanded perturbation sets when giving their formal guarantees. However, their expansions are not based on tolerance $\gamma > 0$.

4.3 Preliminaries

Let $\mathscr{H}$ be a family of binary classifiers $\{h : \mathbb{R}^d \to \{\pm 1\}\}$, and $U = \{U_x \subseteq \mathbb{R}^d : x \in \mathbb{R}^d\}$ any set of robustness regions. We define the robust loss function with respect to U as follows.

Definition 4.3.1. *Let $h \in \mathscr{H}$ be a classifier and $(x,y) \in \mathbb{R}^d \times \{\pm 1\}$ be a labeled point. Then the **robust loss** of h over (x,y), denoted $\ell_U(h,(x,y))$, is defined as*

$$\ell_U(h,(x,y)) = \begin{cases} 1 & \exists x' \in U_x \text{ such that } h(x') \neq y \\ 0 & \text{otherwise.} \end{cases}.$$

That is, h achieves a loss of 0 only if it labels all points in U_x as y.

For a distribution, $\mathscr{D}$ over $\mathbb{R}^d \times \{\pm 1\}$, we let $\ell_U(h,\mathscr{D})$ denote the expected loss h pays over a labeled point drawn from $\mathscr{D}$. That is, $\ell_U(h,\mathscr{D}) = \mathbb{E}_{(x,y)\sim\mathscr{D}}[\ell_U(h,(x,y))]$.

Similarly, for a set of n labeled points, S, we let $\ell_U(h,S)$ denote the average robust loss h pays over S. that is, $\ell_U(h,S) = \frac{1}{n}\sum_{i=1}^{n} \ell_U(h,(x_i,y_i))$.

We will also use $||x-x'||$ to denote the ℓ_2 distance between x and x', and $B(x,r)$ to denote the (closed) ℓ_2 ball centered at x with radius r.

4.3.1 Robust PAC-learning

We now review a natural generalization of PAC learning to the robust setting called robust PAC-learning [9].

Definition 4.3.2. *Let $\mathscr{H}$ be a hypothesis class and U be a set of robustness regions. A learner L **robustly PAC-learns** $(\mathscr{H},U)$ if for every $\varepsilon,\delta > 0$, there exists $m(\varepsilon,\delta)$ such that for all*

$n \geq m(\varepsilon, \delta)$, for all data distributions, $\mathscr{D}$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$,

$$\ell_U(\hat{h}, \mathscr{D}) \leq \min_{h \in \mathscr{H}} \ell_U(h, \mathscr{D}) + \varepsilon,$$

*where $\hat{h} = L(S)$ denotes the classifier in $\mathscr{H}$ outputted by L from training sample S. $m(\varepsilon, \delta)$ is said to be the **sample complexity** of L with respect to $(\mathscr{H}, U)$.*

Algorithms that are able to robustly PAC-learn a pair $(\mathscr{H}, U)$ are the natural robust analogs of standard learning algorithms, and thus an important question is understanding how the sample complexities, $m(\varepsilon, \delta)$, for doing so are bounded.

4.4 Robust Empirical Risk Minimization on Linear Classifiers

Montasser et al. [9] showed that there exist hypothesis classes $\mathscr{H}$ with bounded VC dimension, and robustness regions U, such that proper robust PAC-learning is not possible, meaning no matter how much data one is allowed, there always exists a distribution where the learner will suffer high robust loss.

However, for many practical examples, this does not appear to be the case – for example, [49] showed that when $\mathscr{H}$ is the set of all linear classifiers and U is the set of robustness regions with $U_x = B(x, r)$, the sample complexity of robustly learning with RERM is at most $m(\varepsilon, \delta) = \tilde{O}\left(\frac{d}{\varepsilon^2}\right)$, matching the standard complexity for linear classification.

Motivated by recent interest in more general robustness regions than balls of a fixed radius, we consider the case where $\mathscr{H}$ is a natural hypothesis class, but U is a potentially arbitrary robustness region. That is, we ask the following question: are there examples of natural hypothesis classes for which there exist robustness regions leading to arbitrary high sample complexities?

Unfortunately, the answer turns out to be yes. To show this, we begin by defining the natural hypothesis class of *bounded* linear classifiers.

Definition 4.4.1. *A W-bounded linear classifier, $f : \mathbb{R}^d \to \mathbb{R}^d$, is a linear classifier h whose decision boundary has distance at most W from the origin. That is, there exist $w \in \mathbb{R}^d$ and , $b \in \mathbb{R}$ with $\frac{|b|}{||w||} \leq W$ such that*

$$h(x) = \begin{cases} 1 & \langle w, x \rangle + b \geq 0 \\ -1 & otherwise \end{cases}.$$

We let $\mathscr{H}_W$ denote the class of all W-bounded linear classifiers

The boundedness condition, W, can be thought of as a regularization term which is common during any kind of practical optimization.

We now show that there exist robustness regions, U, for which $(\mathscr{H}_W, U)$ is not robustly PAC-learnable, even in the realizable setting. For convenience, we restate Theorem 4.4.2 from the introduction.

Theorem 4.4.2. *For any $W > 0$ and $d > 1$, there exists a set of robustness regions U over $\mathbb{R}^d$ such that for any learning algorithm L there exists a distribution $\mathscr{D}$ for which the following hold:*

- *$\mathscr{D}$ **is realizable**: There exists $h^* \in \mathscr{H}_W$ such that $\ell_U(h^*, \mathscr{D}) = 0$.*
- *L **has high error**: With probability at least $\frac{1}{7}$ over $S \sim \mathscr{D}^m$, $\ell_U(L(S), \mathscr{D}) > \frac{1}{8}$.*

Theorem 4.4.2 consequently shows that the observations made in [9] hold even over practical hypothesis classes such as (bounded) linear classifiers.

To prove Theorem 4.4.2, we begin with the following critical lemma.

Lemma 4.4.3. *For every $M \in \mathbb{N}$ there exists a family of M subsets of $\mathbb{R}^d$*

$$Z^{(M)} := \left\{ Z_1^{(M)}, Z_2^{(M)}, \ldots, Z_M^{(M)} \right\}$$

satisfying the following conditions:

- *For every $h \in \mathscr{H}_W$, there exists $z \in Z^{(M)}$ such that $h(z) = 1$.*
- *For every $1 \leq i \leq M$, there exists $h_i \in \mathscr{H}_W$ such that $h_i(z) = -1$ for all $z \in \cup_{j \neq i} Z_j^{(M)}$.*
- *The sets $\{Z^{(M)}\}_{M \in \mathbb{N}}$ are mutually disjoint.*

Proof. Let $\{\beta_i\}_{i \in \mathbb{N}} > 0$ be a strictly decreasing sequence of sufficiently small real numbers (that we will specify later). For notational simplicity, fix an $M \in \mathbb{N}$ and write $\beta = \beta_M$ and $W' = (1+\beta)W$. For any $r > 0$, let S_r^{d-1} denote the $(d-1)$-sphere centered at the origin of radius r.

Observe that for any $x \in S_W^{d-1}$, there exists a unique classifier $h \in \mathscr{H}_W$ whose decision boundary is tangent to S_W^{d-1} at x so that $h(x) = 1$. We denote this classifier as h_x. It follows that the set of all points on $S_{W'}^{d-1}$ that h_x classifies as 1 can be easily characterized in terms of x. In particular, by the definition of h_x, it follows from geometry that

$$\left\{z : h_x(z) = 1, z \in S_{W'}^{d-1}\right\} = \left\{z : ||z - (1+\beta)x|| \leq W\sqrt{2\beta(\beta+1)}, z \in S_{W'}^{d-1}\right\}. \tag{4.1}$$

Let $r_\beta = 2W\sqrt{2\beta(\beta+1)}$, and let $z_1, z_2, \ldots, z_{M_\beta}$ denote a a greedy r_β cover of $S_{W'}^{d-1}$, meaning that points are successively selected from $S_{W'}^{d-1}$ until no point with distance strictly greater than r_β from all other points can be selected. Finally, define $Z_i = Z_i^{(M)}$ as the set of elements in $S_{W'}^{d-1}$ with nearest neighbor z_i (ties broken arbitrarily).

We claim that this construction suffices for $M_\beta \geq M$. First, observe that $\lim_{\beta \to 0} r_\beta = 0$, which means that for sufficiently small β that M_β will be arbitrarily large (thus satisfying $M_\beta \geq M$). So select any β for which this hold, and merge enough regions so that we are left with exactly M regions (i.e. set $Z_M = \cup_{i=M}^{M_\beta} Z_i$). Note that we can always choose $0 < \beta < \beta_{M-1}$ since the naturals can be embedded into any interval. We now verify the two stipulations of Lemma 4.4.3.

The first stipulation clearly holds since $\{Z_i\}_{i=1}^M$ partition $S_{W'}^{d-1}$ and every halfspace $h \in \mathscr{H}_W$ intersects the latter by construction.

For the second stipulation, observe that for any i, the ball centered at z_i of radius $\frac{r_\beta}{2}$, $B\left(z_i, \frac{r_\beta}{2}\right)$, does not intersect Z_j for any $i \neq j$. This is because such an intersection would imply by the triangle inequality that $||z_i - z_j|| \leq r_\beta$, which is a contradiction. This observation allows us to find a classifier, h_i, as desired – we set h_i to be the previously defined classifier, $h_{\frac{z_i}{1+\beta}}$. Equation 4.1 implies that the only points in $S^{d-1}_{W'}$ that it will classify as 1 are precisely the points in $B\left(z_i, \frac{r_\beta}{2}\right) \cap S^{d-1}_{W'}$. Since this is a subset of Z_i, the second stipulation is met, as desired.

Finally, it is left to observe that over each choice of M these $Z^{(M)}$ are mutually disjoint. This is true so long as the choices of β themselves are disjoint, since $Z^{(M)}$ lies in the sphere of radius $W(1+\beta_M)$. As noted previously it is easy to see $\{\beta_M\}$ can be chosen in this manner in an inductive fashion. □

We are now sketching a proof for Theorem 4.4.2, with the full proof deferred Appendix D.1.

Proof Sketch: (Theorem 4.4.2)

Our goal is to show that for any $m \in \mathbb{N}$, any learner on m samples must fail with constant probability. Fix any m. The main idea will be to construct a set of robustness regions, $U_{x_1}, U_{x_2}, \ldots, U_{x_{3m}}$ such that any classifier in $\mathscr{H}_W$ will lack robustness on at least m of them. T

Toward this end, set $M = \binom{3m}{m}$, and let $Z^{(M)}_1, Z^{(M)}_2, \ldots, Z^{(M)}_M$ be subsets of $\mathbb{R}^d$ as described by Lemma 4.4.3 (we will drop the superscript in what follows). Let $\mathscr{M}$ denote the set of all subsets of $\{1, \ldots, 3m\}$ with exactly m elements. Associate with each Z_i a unique element of $\mathscr{M}$, thus allowing us to rename our subsets as $\{Z_T : T \in \mathscr{M}\}$. We now define

$$U_{x_i} = \cup_{T: i \in T} Z_T,$$

where x_i is an arbitrary point inside U_{x_i}.

Lemma 4.4.3 that if all x_i are given a label of -1, then any $h \in \mathscr{H}_W$ will label some (for some set T) some $z \in Z_T$ as $+1$, thus causing it to lack robustness on *all* $i \in T$. Conversely, we

see that for any T, there is a classifier $h_T \in \mathscr{H}_W$ that is accurate and robust at all x_i with $i \notin T$.

With these observations, we are now prepared to show that for any learner L, there exists a distribution D for which L has large expected robust loss. To do this, we use a standard lower bound technique found in [62] that was adapted to the robust setting in [9]. The idea will be to pick D to be the uniform distribution over a random subset of $2m$ points in $\{x_1, \ldots, x_{3m}\}$. We will then argue that because L only has access to m points from D, it won't be able to distinguish which subset D corresponds to, and this will lead to a large expected loss. □

As demonstrated in Lemma 4.4.3, the robustness regions U used in our lower bound are combinatorial in nature and unlikely to represent any practical kinds of robustness regions. Nevertheless, our lower bound does show that naturality assumptions on the hypothesis class alone are *not* sufficient for ensuring robust PAC-learnability.

A natural next step would be to fully characterizes pairs $(\mathscr{H}, U)$ for which proper robust PAC-learnability is possible, but we leave this as a direction for future work. We instead turn towards relaxing the requirements of the robust PAC-learning model in order to find algorithms that are able to succeed in the case that $\mathscr{H}$ is natural but U is arbitrary.

4.5 Tolerant PAC learning

Theorem 4.4.2 implies that for complex robustness regions, robust PAC-learning (Definition 4.3.2) is not possible, even when $\mathscr{H}$ is a simple hypothesis class. Thus, robust learning will require other ideas.

One such idea is Tolerant PAC-learning, introduced in Ashtiani et al. [60]. Here, the idea is to relax the goal of robust PAC-learning by introducing a tolerance parameter γ representing the amount of "slack" the learner gets with respect to the robustness regions U. We now expand their definition to arbitrary robustness regions by introducing *perturbed regions*, U^γ, which are defined as follows.

Definition 4.5.1. *Let U be a set of robustness regions and $\gamma > 0$ be a distance. For any point*

$x \in \mathbb{R}^d$, define U_x^γ as the set of all points with distance at most γ from U_x. That is,

$$U_x^\gamma = \{x' : ||x' - U_x|| \leq \gamma\}.$$

*Finally, we let $U^\gamma = \{U_x^\gamma : x \in \mathbb{R}^d\}$ denote the set of **γ-perturbed regions** of U.*

Tolerant PAC-learning is then defined as follows

Definition 4.5.2. *Let $\mathscr{H}$ be a hypothesis class and U a set of robustness regions. A learner L **tolerantly PAC-learns** $(\mathscr{H}, U)$ if for every $\varepsilon, \delta, \gamma > 0$, there exists $m(\varepsilon, \delta, \gamma)$ such that for all $n \geq m(\varepsilon, \delta, \gamma)$, for all data distributions, $\mathscr{D}$, with probability $1 - \delta$ over $S \sim \mathscr{D}^n$,*

$$\ell_U(\hat{h}, \mathscr{D}) \leq \min_{h \in \mathscr{H}} \ell_{U^\gamma}(h, \mathscr{D}) + \varepsilon,$$

*where $\hat{h} = L(S)$ denotes the classifier outputted by L from training sample S. As before, we let $m(\varepsilon, \delta, \gamma)$ denote the **tolerant sample complexity** of L with respect to $(\mathscr{H}, U)$.*

4.5.1 Tolerant RERM oracles

Because our robustness regions, U_x, are arbitrary subsets of $\mathbb{R}^d$, any learning algorithm will require some sort of access to U. We describe this access through an oracle for U.

Ashtiani et al. [60] employs a *sampling oracle* for U which allows the learner to sample points at uniform from the set U_x for any point x. In their setting, U_x is constrained to be a closed ball of known radius centered at x, and consequently the sampling oracle selects points from the uniform distribution over the ball. We say that a robust learner is in the *sampling model* if its only way of interacting with the regions U_x is through a sampling oracle.

In our setting, where U_x can be an arbitrary regions, sampling oracles pose a significant challenge – there exists choices of U for which tolerant PAC learning requires an exponential number of queries to the sampling oracle. We state this as a proposition with the proof in Appendix D.2.

Proposition 4.5.3. *For any $D > 10\gamma > 0$, there exists a hypothesis class $\mathscr{H}$ and a set of robustness regions, U such that the following holds. There exist constants ε and δ, along with a data distribution $\mathscr{D}$, such that for any $n > 0$, any learner L that achieves*

$$\ell_U(L(S), \mathscr{D}) \leq \min_{h \in \mathscr{H}} \ell_{U^\gamma}(h, \mathscr{D}) + \varepsilon$$

with probability at least $1 - \delta$ over $S \sim \mathscr{D}^n$ must make at least $\Omega\left(\left(\frac{D}{\gamma}\right)^d\right)$ sampling oracle calls.

To circumvent this issue, we turn our attention to a different natural oracle first proposed in Montasser et al. [9] that is based on Robust Empirical Risk Minimization (RERM). An RERM oracle, $\mathscr{O}_{U,\mathscr{H}}(S)$, is a function that returns the classifier $h \in \mathscr{H}$ with minimal robust empirical risk over S. That is,

$$\mathscr{O}_{U,\mathscr{H}}(S) = \arg\min_{h \in \mathscr{H}} \ell_U(h, S).$$

In our work, we will assume access to a mild strengthening of this oracle that allows empirical risk minimization over any perturbed robustness region, U^r.

Definition 4.5.4. ***A tolerant RERM-oracle*** *for robustness regions U and hypothesis class $\mathscr{H}$ is a function $\mathscr{O}_{U,\mathscr{H}}(S, r)$ that maps any set of labeled points S and any distance $r > 0$ to the classifier with minimal empirical risk over S with respect to U^r. That is,*

$$\mathscr{O}_{U,\mathscr{H}}(S, r) = \arg\min_{h \in H} \ell_{U^r}(h, S).$$

Observe that in the case that U consists of balls of radius r, a tolerant oracle merely implies we can also minimize empirical risk for balls of larger radii.

4.6 Tolerant PAC learning for Regular Hypothesis Classes

Before presenting our algorithm, we first present a key assumption on our hypothesis class, $\mathscr{H}$, that we refer to as *regularity.*

4.6.1 Regular hypothesis classes

Definition 4.6.1. *We say that a hypothesis class, $\mathscr{H}$ is α-**regular** for $\alpha > 0$ if for all $h \in \mathscr{H}$ and for all $x \in \mathbb{R}^d$, there exists a closed ball B of radius α containing x such that $h(x') = h(x)$ for all $x' \in B$. We also say that $\mathscr{H}$ is **regular** if it is α-regular for some $\alpha > 0$.*

One important example is hypothesis classes with relatively smooth manifolds as decision boundaries. In particular, the parameter α can be tied to the smoothness measure of a manifold known as its *reach*.

Definition 4.6.2. *Let M be a closed manifold embedded in $\mathbb{R}^d$. The **reach** of M is the largest $\alpha > 0$ such that for all $x \in \mathbb{R}^d$, if $||x - M|| \leq \alpha$, then x has a unique nearest neighbor in M.*

This parameter directly translates to regularity.

Proposition 4.6.3. *Let h be a classifier with decision boundary M. Suppose that M is a closed $(d-1)$-dimensional submanifold over $\mathbb{R}^d$ with reach α. Then h is $\alpha/2$-regular.*

Proof. Let $h \in \mathscr{H}$ be a classifier with decision boundary M. Let x be an arbitrary point with $h(x) = y$. We desire to exhibit a ball B of radius $\alpha/2$ containing x for which h is uniformly y.

Let $\rho : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ be the distance function $\rho(x) = ||x - M||$. It is well known that this function is everywhere continuous and has a continuous derivative over $\{x : 0 < \rho(x) < \alpha\}$.

If $\rho(x) > \alpha/2$, then we can simply take $B = B(x, \alpha/2)$ as all points here must be classified as y by the definition of a decision boundary. Thus, assume $\rho(x) \leq \alpha/2$.

Let V be the gradient vector field of ρ defined over $\{x : \rho(x) < \alpha\}$. Since all points in this region have a unique nearest neighbor in M, the gradient has magnitude 1 for all such points,

and the direction is precisely opposite the straight line path from the point's nearest neighbor in M.

Since V is continuous, (and Lipshitz over a bounded region), there exists a unique curve τ starting at x of length $\frac{\alpha}{2}$ that is always tangent to V. It follows that the endpoint of this path, x' must satisfy $\rho(x') = \frac{\alpha}{2} + \rho(x) > \frac{\alpha}{2}$ and $||x - x'|| \leq \frac{\alpha}{2}$. This means that $B = B(x', \frac{\alpha}{2})$ suffices, as desired. □

4.6.2 Our Algorithm

We now give a tolerant PAC learning algorithm called *TolRERM* (Algorithm 4) which assumes access to a tolerant RERM oracle (Definition 4.5.4). *TolRERM* is essentially robust empirically risk minimization with a slight modification: rather than using the original robustness regions, U, we use the perturbed regions, U^r where $0 < r < \gamma$ is chosen at random. *TolRERM*'s performance is given by Theorem 4.1.2, which is restated here for convenience.

Theorem 4.6.4. *Let $\mathscr{H}$ be a regular hypothesis class with VC dimension v, and let U be a set of robustness regions. Then TolRERM tolerantly PAC-learns $(\mathscr{H}, U)$ with tolerant sample complexity, $m(\varepsilon, \delta, \gamma) = O\left(\frac{vd \log \frac{dD}{\varepsilon\gamma\delta}}{\varepsilon^2}\right)$, where D denotes the maximum ℓ_2 diameter of any region, U_x.*

Algorithm 4: $TolRERM(\mathscr{D}, \varepsilon, \delta, \gamma, n)$

1 Sample $r \sim [\frac{\varepsilon\delta\gamma}{7}, \gamma]$ at uniform;
2 Sample $S \sim \mathscr{D}^n$;
3 Output $\hat{h} = \mathscr{O}_{U,\mathscr{H}}(S, r)$;

Since the set of bounded linear classifiers, $\mathscr{H}_W$ (Definition 4.4.1) is clearly regular and has VC dimension $O(d)$, Theorem 4.1.2 immediately implies the following corollary.

Corollary 4.6.5. *For any set of robustness regions, U, TolRERM tolerantly PAC-learns $(\mathscr{H}_W, U)$ with tolerant sample complexity $m(\varepsilon, \delta, \gamma) = O\left(\frac{d^2 \log \frac{dD}{\varepsilon\gamma\delta}}{\varepsilon^2}\right)$, where D denotes the maximum ℓ_2 diameter of any robustness region, U_x.*

TolRERM matches the sample complexities for linear classifiers found in [9] and [60]. However, it enjoys the advantage of being simpler (as it is essentially an empirical risk minimization algorithm) and a *proper* learning algorithm (as it outputs a linear classifier).

Beyond regular hypothesis classes:

It turns out that Algorithm 4 has bounded sample complexity for *any* hypothesis class with finite robust VC-dimension for balls (see Appendix D.4 for a full description). Thus, Algorithm 4 can alternatively be thought of as a reduction from the sample complexity for learning robust classifiers over arbitrary robustness regions to the sample complexity for balls of fixed radii. This is expressed in the following result (proved in Appendix D.4).

Theorem 4.6.6. *Let $\mathscr{H}$ be any hypothesis class with maximal adversarial VC dimension v_{ball}, and let U be any set of robustness regions. Then TolRERM tolerantly PAC-learns $(\mathscr{H}, U)$ with tolerant sample complexity $m(\varepsilon, \delta, \gamma) = O\left(\frac{v_{ball} d \log \frac{dD}{\varepsilon\gamma\delta}}{\varepsilon^2}\right)$, where D denotes the maximum ℓ_2 diameter of any robustness region, U_x.*

4.6.3 Proof of Theorem 4.1.2

We begin by showing that randomly choosing r allows the optimal empirical loss U^r to change relatively smoothly with respect to r.

Lemma 4.6.7. *For $r \in [0, \gamma]$, let $OPT_S^r = \min_{h \in H} \ell_{U^r}(h, S)$. Then with probability at least $1 - \frac{\delta}{2}$ over $r \sim [\frac{\varepsilon\delta\gamma}{7}, \gamma]$, $OPT_S^r \leq OPT_S^{r - \frac{\varepsilon\delta\gamma}{7}} + \frac{\varepsilon}{3}$.*

Proof. Let $\alpha = \frac{\varepsilon\delta\gamma}{7}$. Our goal is to show that $OPT_S^r - OPT_S^{r-\alpha}$ is likely to be small. Our strategy is to bound the expected value of $OPT_S^r - OPT_S^{r-\alpha}$ and then apply Markov's inequality. As a technical note, the function $r \mapsto OPT_S^r$ is monotonic and bounded, and consequently

measurable, which ensures that our expectations are well defined. To this end, we have,

$$\begin{aligned}
\mathbb{E}[OPT_S^r - OPT_S^{r-\alpha}] &= \mathbb{E}[OPT_S^r] - \mathbb{E}[OPT_S^{r-\alpha}] \\
&= \frac{1}{\gamma-\alpha}\left(\int_\alpha^\gamma OPT_S^r dr - \int_\alpha^\gamma OPT_S^{r-\alpha} dr\right) \\
&= \frac{1}{\gamma-\alpha}\left(\int_\alpha^\gamma OPT_S^r dr - \int_0^{\gamma-\alpha} OPT_S^r dr\right) \\
&= \frac{1}{\gamma-\alpha}\left(\int_{\gamma-\alpha}^\gamma OPT_S^r dr - \int_0^\alpha OPT_S^r dr\right) \\
&\leq \frac{\alpha}{\gamma-\alpha} = \frac{\delta\varepsilon\gamma}{7\gamma-\delta\varepsilon\gamma} \leq \frac{\delta\varepsilon}{6},
\end{aligned}$$

since $\varepsilon, \delta \leq 1$. Applying Markov's inequality, with probability at least $1-\frac{\delta}{2}$, $OPT_S^r - OPT_S^{r-\alpha} \leq \frac{\varepsilon}{3}$. $\square$.

Next, we construct a set of robustness regions V^r that have similar robust loss to U^r and are also finite.

Lemma 4.6.8. *Suppose that $\mathscr{H}$ is γ-regular. For all $r \in [\frac{\varepsilon\delta\gamma}{7}, \gamma]$, there exists a set of robustness regions $V^r = \{V_x^r : x \in \mathbb{R}^d\}$ satisfying the following two properties.*

1. *$|V_x^r| = O\left(\left(\frac{D}{\varepsilon\delta\gamma}\right)^d\right)$, where D denotes the maximum diameter of U_x.*
2. *Let $\alpha = \frac{\varepsilon\delta\gamma}{7}$. For all labeled points (x,y) and for all classifiers $h \in \mathscr{H}$,*

$$\ell_{U^{r-\alpha}}(h,(x,y)) \leq \ell_{V^r}(h,(x,y)) \leq \ell_{U^r}(h,(x,y)).$$

Proof. For any $x \in \mathbb{R}^d$, we will show how to construct V_x so that it satisfies the two conditions above.

Observe that U_x^r is closed and bounded as it is a union of closed balls of radius r. Since each U_x has diameter at most D, this means that U_x^r is compact. Thus, there exists a finite set of balls of radius $\alpha/2$ that cover U_x^r. Note that these balls are *not* necessarily contained within U_x^r – only that U_x^r is a subset of their union. Let C_x denote the set of all centers of the smallest such

cover. We claims that $V_x = C_x \cap U_x^r$ suffices.

First, $|C_x| \leq O\left((\frac{D}{\alpha})^d\right)$ because any ball of diameter D can be covered by $O\left((\frac{D}{\alpha})^d\right)$ balls of radius $\alpha/2$, and U_x^r is a subset of a ball of diameter $D+2r$. This implies that the first condition holds.

Second, pick any labeled point (x,y) and any classifier $h \in \mathscr{H}$. If $\ell_{V^r}(h,(x,y)) = 1$, then we immediately have $\ell_{U^r}(h,(x,y) = 1$ since $V^r \subseteq U^r$. This implies that $\ell_{V^r}(h,(x,y)) \leq \ell_{U^r}(h,(x,y)$ giving the second half of the second condition.

If $\ell_{U^{r-\alpha}}(h,(x,y)) = 1$, then there exists $x' \in U_x^{r-\alpha}$ such that $h(x') \neq y$. It follows that since h is γ-regular, h must also be α-regular (as $\alpha < \gamma$). This means that there exists a ball B of radius $\alpha/2$ containing x' such that h does not output y for any point in B.

By the triangle inequality, $B \subseteq U_x^r$, and since C_x covers U_x^r, it follows that there exists $x^* \in C_x \cap B$. By definition, this also means $x^* \in V_x^r$. However, by the definition of B, we must have $h(x^*) \neq y$, and this means that $\ell_{V_x^r}(h,(x,y)) = 1$. Since (x,y) was arbitrary, this proves the second half of the second condition. □

We are now prepared to prove Theorem 4.1.2.

Proof. (**Theorem 4.1.2**) Let $\alpha = \frac{\varepsilon\delta\gamma}{7}$. For all $s > 0$, let $h^s \in \mathscr{H}$ denote any fixed choice of classifier with minimal empirical loss with respect to U^s. That is, $h^s = \arg\min_{h\in\mathscr{H}} \ell_{U^s}(h,S)$. Then by Lemma 4.6.7, with probability at least $1 - \frac{\delta}{2}$ over $r \sim [\alpha,\gamma]$,

$$\ell_{U^r}(h^r,S) \leq \ell_{U^{r-\alpha}}(h^{r-\alpha},S) + \frac{\varepsilon}{3}. \tag{4.2}$$

Next, let V^r be as defined in Lemma 4.6.8 and suppose that γ is small enough so that $\mathscr{H}$ is γ-regular (this must occur since $\mathscr{H}$ is regular by assumption). By the second condition in the lemma, it follows that for all $h \in H$:

$$\ell_{U^{r-\alpha}}(h,\mathscr{D}) \leq \ell_{V^r}(h,\mathscr{D}) \leq \ell_{U^r}(h,\mathscr{D}), \tag{4.3}$$

$$\ell_{U^{r-\alpha}}(h,S) \leq \ell_{V^r}(h,S) \leq \ell_{U^r}(h,S). \tag{4.4}$$

Next, since $|V_x| = O\left(\left(\frac{D}{\varepsilon\delta\gamma}\right)^d\right)$, Proposition D.3.3 (proved in the Appendix D.3) implies that the Robust VC dimension of $\mathscr{H}$ with respect to V_x is at most $O\left(vd\log\frac{Dv}{\varepsilon\delta\gamma}\right)$, where v denotes the VC dimension of $\mathscr{H}$.

Because S is independent from r, there exists an absolute constant C such that if $n \geq C\frac{vd\log\frac{Dv}{\varepsilon\delta\gamma}+\log\frac{1}{\delta}}{\varepsilon^2}$, then classical connections with uniform convergence [63] imply that with probability at least $1-\frac{\delta}{2}$ over $S\sim\mathscr{D}^n$, for all $h\in\mathscr{H}$,

$$|\ell_{V^r}(h,S) - \ell_{V^r}(h,\mathscr{D})| \leq \frac{\varepsilon}{3}. \tag{4.5}$$

Applying a union bound, we see that for $n = \Omega\left(\frac{vd\log\frac{dD}{\varepsilon\gamma\delta}}{\varepsilon^2}\right)$, with probability at least $1-\delta$ over $r\sim[\alpha,\gamma]$ and $S\sim\mathscr{D}^n$, Equations 4.2, 4.3, 4.4, and 4.5 simultaneously hold. Thus it suffices to show that under these assumptions, any $\hat{h}\in H$ minimizing the robust empirical risk of S under U^r satisfies $\ell_U(\hat{h},\mathscr{D}) \leq \min_{h\in\mathscr{H}}\ell_{U^\gamma}(h^\gamma,\mathscr{D})+\varepsilon$, as this will imply that $m(\varepsilon,\delta,\gamma) = O\left(\frac{vd\log\frac{dD}{\varepsilon\gamma\delta}}{\varepsilon^2}\right)$ as desired.

To do so, we use a series of manipulations applying Equations 4.2 and 4.5. For convenience, we let $h^* = \arg\min_{h\in\mathscr{H}}\ell_{U^\gamma}(h,\mathscr{D})$.

Since $U\subset U^r$ and ℓ_{U^r} is bounded by ℓ_{V^r} (Equation 4.3), we have that

$$\ell_U(h^r,\mathscr{D}) \leq \ell_{U^r}(h^r,\mathscr{D}) \leq \ell_{V^r}(h^r,\mathscr{D}),$$

and further by Equations 4.5 and 4.4 that

$$\ell_{V^r}(h^r,\mathscr{D}) \leq \ell_{V^r}(h^r,S)+\frac{\varepsilon}{3} \leq \ell_{U^r}(h^r,S)+\frac{\varepsilon}{3}.$$

Since h^s is defined as the classifier of lowest empirical risk over U^s, it follows from Equation 4.2

and this definition that

$$\ell_{U^r}(h^r, S) \leq \ell_{U^{r-\alpha}}(h^{r-\alpha}, S) + \frac{\varepsilon}{3} \leq \ell_{U^{r-\alpha}}(h^*, S) + \frac{\varepsilon}{3}.$$

Applying the same trick we did earlier with Equations 4.4 (bounding $U^{r-\alpha}$ with V^r) and 4.5 (uniform convergence of the loss over V^r), we have

$$\ell_{U^{r-\alpha}}(h^*, S) \leq \ell_{V^r}(h^*, S) \leq \ell_{V^r}(h^*, \mathscr{D}) + \frac{\varepsilon}{3}.$$

Finally, applying Equation 4.3 to bound the loss over V^r with the loss over U^r and then noting that $r \leq \gamma$, we have that

$$\ell_{V^r}(h^*, \mathscr{D}) \leq \ell_{U^r}(h^*, \mathscr{D}) \leq \ell_{U^\gamma}(h^*, \mathscr{D}).$$

Combining all of our observations with the transitive property, it follows that

$$\ell_U(h^r, \mathscr{D}) \leq \ell_{U^\gamma}(h^*, \mathscr{D}) + \varepsilon.$$

Finally, since this holds for any choice of h^r minimizing $\ell_{U^r}(h^r, S)$, it holds for the particular choice of the Tolerant RERM oracle which completes the result. □

Chapter 5

Data-Copying in Generative Models: A Formal Framework

5.1 Introduction

Deep generative models have shown impressive performance. However, given how large, diverse, and uncurated their training sets are, a big question is whether, how often, and how closely they are memorizing their training data. This question has been of considerable interest in generative modeling [64, 65] as well as supervised learning [66, 67]. However, a clean and formal definition of memorization that captures the numerous complex aspects of the problem, particularly in the context of continuous data such as images, has largely been elusive.

For generative models, [3] proposed a formal definition of memorization called "data-copying", and showed that it was orthogonal to various prior notions of overfitting such as mode collapse [68], mode dropping [69], and precision-recall [70]. Specifically, their definition looks at three datasets – a training set, a set of generated example, and an independent test set. Data-copying happens when the training points are considerably closer on average to the generated data points than to an independently drawn test sample. Otherwise, if the training points are further on average to the generated points than test, then there is underfitting. They proposed a three sample test to detect this kind of data-copying, and empirically showed that their test had good performance.

However, despite its practical success, this method may not capture even blatant cases of

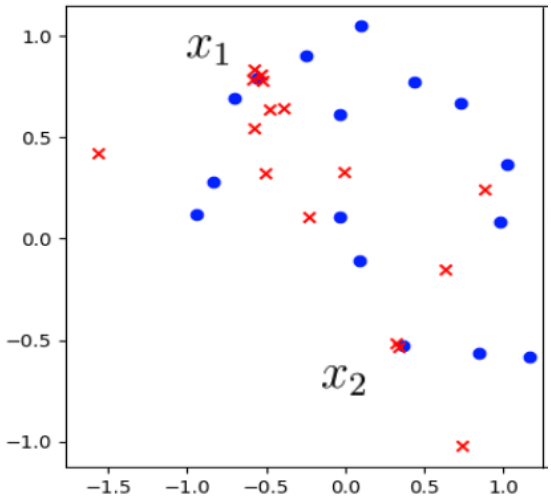


Figure 5.1. In this figure, the blue points are sampled from the halfmoons dataset (with Gaussian noise). The red points are sampled from a generated distribution that is a mixture of (40 %) blatant data copier (that outputs a random subset of the training set), and (60 %) a noisy underfit version of halfmoons. Although the generated distribution is clearly doing some form of copying at points x_1 and x_2, detecting this is challenging because of the canceling effect of the underfit points.

memorization. To see this, consider the example illustrated in Figure 5.1, in which a generated model for the halfmoons dataset outputs one of its training points with probability 0.4, and otherwise outputs a random point from an underfit distribution. When the test of [3] is applied to this distribution, it is unable to detect any form of data copying; the generated samples drawn from the underfit distribution are sufficient to cancel out the effect of the memorized examples. Nevertheless, this generative model is clearly an egregious memorizer as shown in points x_1 and x_2 of Figure 5.1.

This example suggests a notion of *point-wise* data copying, where a model q can be thought of as copying a given training point x. Such a notion would be able to detect q's behavior nearby x_1 and x_2 regardless of the confounding samples that appear at a global level. This stands in contrast to the more global distance based approach taken in Meehan et. al. which is unable to detect such instances. Motivated by this, we propose an alternative point-by-point approach to defining data-copying.

We say that a generative model q data-copies an individual training point, x, if it has an

unusually high concentration in a small area centered at x. Intuitively, this implies q is highly likely to output examples that are very similar to x. In the example above, this definition would flag q as copying x_1 and x_2.

To parlay this definition into a global measure of data-copying, we define the overall *data-copying rate* as the total fraction of examples from q that are copied from some training example. In the example above, this rate is 40%, as this is the fraction of examples that are blatant copies of the training data.

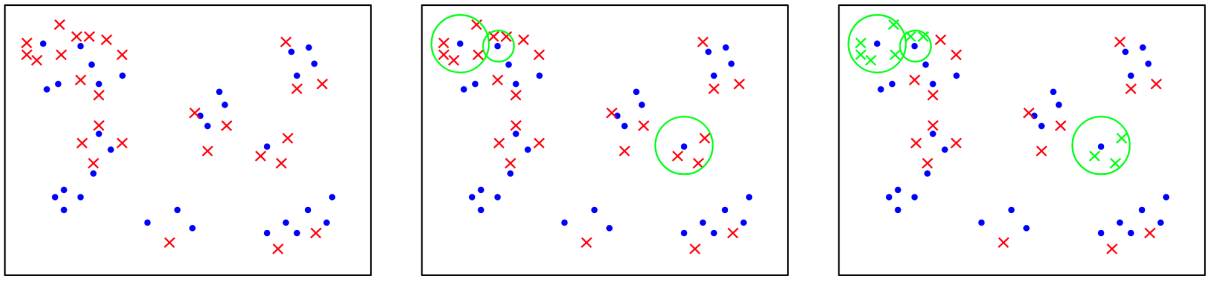


Figure 5.2. In the three panels above, the blue points are a training sample from p, and the red points are generated examples from q. In the middle panel, we highlight in green regions that are defined to be *data-copying regions*, as q overrepresents them with comparison to p. In the third panel, we then color all points from q that are considered to be copied green.

Next, we consider how to detect data-copying according to this definition. To this end, we provide an algorithm, Data_Copy_Detect, that outputs an estimate for the overall data-copying rate. We then show that under a natural smoothness assumption on the data distribution, which we call *regularity*, Data_Copy_Detect is able to guarantee an accurate estimate of the total data-copying rate. We then give an upper bound on the amount of data needed for doing so.

We complement our algorithm with a lower bound on the minimum amount of a data needed for data-copying detection. Our lower bound also implies that some sort of smoothness condition (such as regularity) is necessary for guaranteed data-copying detection; otherwise, the required amount of data can be driven arbitrarily high.

5.1.1 Related Work

Recently, understanding failure modes for generative models has been an important growing body of work e.g. [71, 72, 70]. However, much of this work has been focused on other forms of overfitting, such as mode dropping or mode collapse.

A more related notion of overfitting is *memorization* [64, 65, 73], in which a model outputs exact copies of its training data. This has been studied in both supervised [66, 67] and unsupervised [74, 75] contexts. Memorization has also been considered in language generation models [76].

The first work to explicitly consider the more general notion of *data-copying* is [3], which gives a three sample test for data-copy detection. We include an empirical comparison between our methods in Section 5.5.2, where we demonstrate that ours is able to capture certain forms of data-copying that theirs is not.

Finally, we note that this work focuses on detecting natural forms of memorization or data-copying, that likely arises out of poor generalization, and is not concerned with detecting *adversarial* memorization or prompting, such as in [77], that are designed to obtain sensitive information about the training set. This is reflected in our definition and detection algorithm which look at the specific generative model, and not the algorithm that trains it. Perhaps the best approach to prevent adversarial memorization is training the model with differential privacy [78], which ensures that the model does not change much when one training sample changes. However such solutions come at an utility cost.

5.2 A Formal Definition of Data-Copying

We begin with the following question: what does it mean for a generated distribution q to copy a single training example x? Intuitively, this means that q is guilty of overfitting x in some way, and consequently produces examples that are very similar to it.

However, determining what constitutes a 'very similar' generated example must be done

contextually. Otherwise the original data distribution, p, may itself be considered a copier, as it will output points nearby x with some frequency depending on its density at x. Thus, we posit that q data copies training point x if it has a significantly higher concentration nearby x than p does. We express this in the following definition.

Definition 5.2.1. *Let p be a data distribution, $S \sim p^n$ a training sample, and q be a generated distribution trained on S. Let $x \in S$ be a training point, and let $\lambda > 1$ and $0 < \gamma < 1$ be constants. A generated example $x' \sim q$ is said to be a (λ, γ)-**copy** of x if there exists a ball B centered at x (i.e. $\{x' : ||x' - x|| \leq r\}$) such that following hold:*

- $x' \in B$.
- $q(B) \geq \lambda p(B)$
- $p(B) \leq \gamma$

Here $q(B)$ and $p(B)$ denote the probability mass assigned to B by p and q respectively.

The parameters λ and γ are user chosen parameters that characterize data-copying. λ represents the rate at which q must overrepresent points close to x, with higher values of λ corresponding to more egregious examples of data-copying. γ represents the maximum size (by probability mass) of a region that is considered to be data-copying – the ball B represents all points that are "copies" of x. Together, λ and γ serve as practitioner controlled knobs that characterize data-copying about x.

Our definition is illustrated in Figure 5.2 – the training data is shown in blue, and generated samples are shown in red. For each training point, we highlight a region (in green) about that point in which the red density is much higher than the blue density, thus constituting data-copying. The intuition for this is that the red points within any ball can be thought of as "copies" of the blue point centered in the ball.

Having defined data-copying with respect to a single training example, we can naturally extend this notion to the entire training dataset. We say that $x' \sim q$ is copied from training set S

if x' is a (λ, γ)-copy of some training example $x \in S$. We then define the *data-copy rate* of q as the fraction of examples it generates that are copied from S. Formally, we have the following:

Definition 5.2.2. *Let p, S, q, λ, and γ be as defined in Definition 5.2.1. Then the **data-copy rate**, $cr(q, \lambda, \gamma)$ of q (with respect to p, S) is the fraction of examples from q that are (λ, γ)-copied. That is,*

$$cr(q, \lambda, \gamma) = \Pr_{x' \sim q}[q \ (\lambda, \gamma)\text{-copies } x'].$$

In cases where λ, γ are fixed, we use $cr_q = cr(q, \lambda, \gamma)$ to denote the data-copy rate.

Despite its seeming global nature, cr_q is simply an aggregation of the point by point data-copying done by q over its entire training set. As we will later see, estimating cr_q is often reduced to determining which subset of the training data q copies.

5.2.1 Examples of data-copying

We now give several examples illustrating our definitions. In all cases, we let p be a data distribution, S, a training sample from p, and q, a generated distribution that is trained over S.

The uniform distribution over S:

In this example, q is an egregious data copier that memorizes its training set and randomly outputs a training point. This can be considered as the canonical *worst* data copier. This is reflected in the value of cr_q – if p is a continuous distribution with finite probability density, then for any $x \in S$, there exists a ball B centered at x for which $q(B) >> p(B)$. It follows that q (λ, γ)-copies x for all $x \in S$ which implies that $cr_q = 1$.

The perfect generative model: $q = p$:

In this case, $q(B) = p(B)$ for all balls, B, which implies that q does not perform any data-copying (Definition 5.2.1). It follows that $cr_q = 0$, matching the intuition that q does not data-copy at all.

Kernel Density Estimators:

Finally, we consider a more general situation, where q is trained by a *kernel density estimator* (KDE) over $S \sim p^n$. Recall that a kernel density estimator outputs a generated distribution, q, with pdf defined by

$$q(x) = \frac{1}{n\sigma_n} \sum_{x_i \in S} K\left(\frac{x - x_i}{\sigma_n}\right).$$

Here, K is a kernel similarity function, and σ_n is the bandwidth parameter. It is known that for $\sigma_n = O(n^{-1/5})$, q converges towards p for sufficiently well behaved probability distributions.

Despite this guarantee, KDEs intuitively appear to perform some form of data-copying – after all they implicitly include each training point in memory as it forms a portion of their outputted pdf. However, recall that our main focus is in understanding *overfitting* due to data-copying. That is, we view data-copying as a function of the outputted pdf, q, and not of the training algorithm used.

To this end, for KDEs the question of data-copying reduces to the question of whether q overrepresents areas around its training points. As one would expect, this occurs *before* we reach the large sample limit. This is expressed in the following theorem.

Theorem 5.2.3. *Let $1 < \lambda$ and $\gamma > 0$. Let σ_n be a sequence of bandwidths and K be any regular kernel function. For any $n > 0$ there exists a probability distribution π with full support over $\mathbb{R}^d$ such that with probability at least $\frac{1}{3}$ over $S \sim \pi^n$, a KDE trained with bandwidth σ_n and kernel function K has data-copy rate $cr_q \geq \frac{1}{10}$.*

This theorem completes the picture for KDEs with regards to data-copying – when n is too low, it is possible for the KDE to have a significant amount of data-copying, but as n continues to grow, this is eventually smoothed out.

The Halfmoons dataset

Returning to the example given in Figure 5.1, observe that our definition exactly captures the notion of data-copying that occurs at points x_1 and x_2. For even strict choices of λ and γ, Definition 5.2.1 indicates that the red distribution copies both x_1 and x_2. Furthermore, the data-copy rate, cr_q, is 40% by construction, as this is the proportion of points that are outputted nearby x_1 and x_2.

5.2.2 Limitations of our definition

Definition 5.2.1 implicitly assumes that the goal of the generator is to output a distribution q that approaches p in a mathematical sense; a perfect generator would output q so that $q(M) = p(M)$ for all measurable sets. In particular, instances where q outputs examples that are far away from the training data are considered completely irrelevant in our definition.

This restriction prevents our definition from capturing instances in which q memorizes its training data and then applies some sort of transformation to it. For example, consider an image generator that applies a color filter to its training data. This would not be considered a data-copier as its output would be quite far from the training data in pixel space. Nevertheless, such a generated distribution can be very reasonably considered as an egregious data copier, and a cursory investigation between its training data and its outputs would reveal as much.

The key difference in this example is that the generative algorithm is no longer trying to closely approximate p with q – it is rather trying to do so in some kind of transformed space. Capturing such interactions is beyond the scope of our paper, and we firmly restrict ourselves to the case where a generator is evaluated based on how close q is to p with respect to their measures over the input space.

5.3 Detecting data-copying

Having defined cr_q, we now turn our attention towards *estimating it.* To formalize this problem, we will require a few definitions. We begin by defining a generative algorithm.

Definition 5.3.1. *A **generative algorithm**, A, is a potentially randomized algorithm that outputs a distribution q over $\mathbb{R}^d$ given an input of training points, $S \subset \mathbb{R}^d$. We denote this relationship by $q \sim A(S)$.*

This paradigm captures most typical generative algorithms including both non-parametric methods such as KDEs and parametric methods such as variational autoencoders.

As an important distinction, in this work we define data-copying as a property of the generated distribution, q, rather than the generative algorithm, A. This is reflected in our definition which is given solely with respect to q, S, and p. For the purposes of this paper, A can be considered an arbitrary process that takes S and outputs a distribution q. We include it in our definitions to emphasize that while S is an i.i.d sample from p, it is *not* independent from q.

Next, we define a *data-copying detector* as an algorithm that estimates cr_q based on access to the training sample, S, along with the ability to draw any number of samples from q. The latter assumption is quite typical as sampling from q is a purely computational operation. We do not assume any access to p beyond the training sample S. Formally, we have the following definition.

Definition 5.3.2. *A **data-copying detector** is an algorithm D that takes as input a training sample, $S \sim p^n$, and access to a sampling oracle for $q \sim A(S)$ (where A is an arbitrary generative algorithm). D then outputs an estimate, $D(S,q) = \hat{cr}_q$, for the data-copy rate of q.*

Naturally, we assume D has access to $\lambda, \gamma > 0$ (as these are practitioner chosen values), and by convention don't include λ, γ as formal inputs into D.

The goal of a data-copying detector is to provide accurate estimates for cr_q. However, the precise definition of cr_q poses an issue: data-copy rates for varying values of λ and γ can vastly differ. This is because λ, γ act as thresholds with everything above the threshold being counted, and everything below it being discarded. Since λ, γ cannot be perfectly accounted for, we will require some tolerance in dealing with them. This motivates the following.

Definition 5.3.3. *Let $0 < \varepsilon$ be a tolerance parameter. Then the **approximate data-copy rates**, $cr_q^{-\varepsilon}$ and cr_q^{ε}, are defined as the values of cr_q when the parameters (λ, γ) are shifted by a factor of $(1+\varepsilon)$ to respectively decrease and increase the copy rate. That is,*

$$cr_q^{-\varepsilon} = cr\left(q, \lambda(1+\varepsilon), \gamma(1+\varepsilon)^{-1}\right),$$

$$cr_q^{\varepsilon} = cr\left(q, \lambda(1+\varepsilon)^{-1}, \gamma(1+\varepsilon)\right).$$

The shifts in λ and γ are chosen as above because increasing λ and decreasing γ both reduce cr_q seeing as both result in more restrictive conditions for what qualifies as data-copying. Conversely, decreasing λ and increasing γ has the opposite effect. It follows that

$$cr_q^{-\varepsilon} \leq cr_q \leq cr_q^{\varepsilon},$$

meaning that $cr_q^{-\varepsilon}$ and cr_q^{ε} are lower and upper bounds on cr_q.

In the context of data-copying detection, the goal is now to estimate cr_q in comparison to $cr_q^{\pm\varepsilon}$. We formalize this by defining *sample complexity* of a data-copying detector as the amount of data needed for accurate estimation of cr_q.

Definition 5.3.4. *Let D be a data-copying detector and p be a data distribution. Let $\varepsilon, \delta > 0$ be standard tolerance parameters. Then D has **sample complexity**, $m_p(\varepsilon, \delta)$, with respect to p if for all $n \geq m_p(\varepsilon, \delta)$, $\lambda > 1$, $0 < \gamma < 1$, and generative algorithms A, with probability at least $1 - \delta$ over $S \sim p^n$ and $q \sim A(S)$,*

$$cr_q^{-\varepsilon} - \varepsilon \leq D(S, q) \leq cr_q^{\varepsilon} + \varepsilon.$$

Here the parameter ε takes on a somewhat expanded as it is both used to additively bound our estimation of cr_q and to multiplicatively bound λ and γ.

Observe that there is no mention of the number of calls that D makes to its sampling

oracle for q. This is because samples from q are viewed as *purely computational*, as they don't require any natural data source. In most cases, q is simply some type of generative model (such as a VAE or a GAN), and thus sampling from q is a matter of running the corresponding neural network.

5.4 Regular Distributions

Our definition of data-copying (Definition 5.2.1) motivates a straightforward point by point method for data-copying detection, in which for every training point, x_i, we compute the largest ball B_i centered at x_i for which $q(B_i) \geq \lambda\, p(B_i)$ and $p(B_i) \leq \gamma$. Assuming we compute these balls accurately, we can then query samples from q to estimate the total rate at which q outputs within those balls, giving us our estimate of cr_q.

The key ingredient necessary for this idea to work is to be able to reliably estimate the masses, $q(B)$ and $p(B)$ for any ball in $\mathbb{R}^d$. The standard approach to doing this is through *uniform convergence*, in which large samples of points are drawn from p and q (in p's case we use S), and then the mass of a ball is estimated by counting the proportion of sampled points within it. For balls with a sufficient number of points (typically $O(d \log n)$), standard uniform convergence arguments show that these estimates are reliable.

However, this method has a major pitfall for our purpose – in most cases the balls B_i will be very small because data-copying intrinsically deals with points that are very close to a given training point. While one might hope that we can simply ignore all balls below a certain threshold, this does not work either, as the sheer number of balls being considered means that their union could be highly non-trivial.

To circumvent this issue, we will introduce an interpolation technique that estimates the probability mass of a small ball by scaling down the mass of a sufficiently large ball with the same center. While obtaining a general guarantee is impossible – there exist pathological distributions that drastically change their behavior at small scales – it turns out there is a relatively natural

condition under which such interpolation will work. We refer to this condition as *regularity,* which is defined as follows.

Definition 5.4.1. *Let $k > 0$ be an integer. A probability distribution p is k-**regular** the following holds. For all $\varepsilon > 0$, there exists a constant $0 < p_\varepsilon \leq 1$ such that for all x in the support of p, if $0 < s < r$ satisfies that $p(B(x,r)) \leq p_\varepsilon$, then*

$$\left(1+\frac{\varepsilon}{3}\right)^{-1} \frac{r^k}{s^k} \leq \frac{p(B(x,r))}{p(B(x,s))} \leq \left(1+\frac{\varepsilon}{3}\right) \frac{r^k}{s^k}.$$

*Finally, a distribution is **regular** if it is k-regular for some integer $k > 0$.*

Here we let $B(x,r) = \{x' : ||x - x'|| \leq r\}$ denote the closed ℓ_2 ball centered at x with radius r.

The main intuition for a k-regular distribution is that at a sufficiently small scale, its probability mass scales with distance according to a power law, determined by k. The parameter k dictates how the probability density behaves with respect to the distance scale. In most common examples, k will equal the *intrinsic dimension* of p.

As a technical note, we use an error factor of $\frac{\varepsilon}{3}$ instead of ε for technical details that enable cleaner statements and proofs in our results (presented later).

5.4.1 Distributions with Manifold Support

We now give an important class of k-regular distributions.

Proposition 5.4.2. *Let p be a probability distribution with support precisely equal to a compact k dimensional sub-manifold (with or without boundary) of $\mathbb{R}^d$, M. Additionally, suppose that p has a continuous density function over M. Then it follows that p is k-regular.*

Proposition 5.4.2 implies that most data distributions that adhere to some sort of manifold-hypothesis will also exhibit regularity, with the regularity constant, k, being the intrinsic dimension of the manifold.

5.4.2 Estimation over regular distributions

We now turn our attention towards designing estimation algorithms over regular distributions, with our main goal being to estimate the probability mass of arbitrarily small balls. We begin by first addressing a slight technical detail – although the data distribution p may be regular, this does not necessarily mean that the regularity constant, k, is known. Knowledge of k is crucial because it determines how to properly interpolate probability masses from large radius balls to smaller ones.

Luckily, estimating k turns out to be an extremely well studied task, as for most probability distributions, k is a measure of the *intrinsic dimension*. Because there is a wide body of literature in this topic, we will assume from this point that k has been correctly estimated from S using any known algorithm for doing so (for example [79]). Nevertheless, for completeness, we provide an algorithm with provable guarantees for estimating k (along with a corresponding bound on the amount of needed data) in Appendix E.2.

We now return to the problem of $p(B(x,r))$ for a small value of r, and present an algorithm, $Est(x,r,S)$ (Algorithm 5), that estimates $p(B(x,r))$ from an i.i.d sample $S \sim p^n$.

Algorithm 5: $Est(x,r,S)$

$n \leftarrow |S|$
$b \leftarrow O\left(\frac{d \ln \frac{n}{\delta}}{\varepsilon^2}\right)$
$r_* = \min\{s > 0, |S \cap B(x,s)| = b\}$.
if $r_* > r$ **then**
    Return $\frac{br^k}{nr_*^k}$
else
    Return $\frac{|T \cap B(x,r)|}{n}$

Est uses two ideas: first, it leverages standard uniform convergence results to estimate the probability mass of all balls that contain a sufficient number of training examples. This is what leads to the specific value of b that is chosen. Second, it estimates the mass of smaller balls by interpolating from its estimates from larger balls. The k-regularity assumption is crucial for

this second step as it is the basis on which such interpolation is done.

Est has the following performance guarantee, which follows from standard uniform convergence bounds and the definition of k-regularity.

Proposition 5.4.3. *Let p be a regular distribution, and let $\varepsilon > 0$ be arbitrary. Then if $n = O\left(\frac{d \ln \frac{d}{\delta \varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}\right)$ with probability at least $1-\delta$ over $S \sim p^n$, for all $x \in \mathbb{R}^d$ and $r > 0$,*

$$\left(1+\frac{\varepsilon}{2}\right)^{-1} \leq \frac{Est(x,r,S)}{p(B(x,r))} \leq \left(1+\frac{\varepsilon}{2}\right).$$

5.5 A Data-copy detecting algorithm

Algorithm 6: $DataCopyDetect(S,q,m)$

$m \leftarrow O\left(\frac{dn^2 \ln \frac{nd}{\delta\varepsilon}}{\varepsilon^4}\right)$
Sample $T \sim q^m$
$\{x_1, x_2, \ldots, x_n\} \leftarrow S$
$\{z_1, z_2, \ldots, z_m\} \leftarrow T$
for $i = 1, \ldots, n$ **do**
    Let $p_i(r)$ denote $Est(x_i, r, S)$
    Let $q_i(r)$ denote $\frac{|B(x_i,r) \cap T|}{m}$
    $radii \leftarrow \{||z - x_i|| : z \in T\} \cup \{0\}$
    $radii \leftarrow \{r : p_i(r) \leq \gamma, r \in radii\}$
    $r_i^* \leftarrow \max\{r : q_i(r) \geq \lambda p_i(r), r \in radii\}$
end for
Sample $U \sim q^{20/\varepsilon^2}$
$V \leftarrow U \cap (\bigcup_{i=1}^n B(x_i, r_i^*))$
Return $\frac{|V|}{|U|}$.

We now now leverage our subroutine, *Est*, to construct a data-copying detector, *Data_Copy_Detect* (Algorithm 6), that has bounded sample complexity when p is a regular distribution. Like all data-copying detectors (Definition 5.3.2), *Data_Copy_Detect* takes as input the training sample S, along with the ability to sample from a generated distribution q that is trained from S. It then

performs the following steps:

1. (line 1) Draw an i.i.d sample of $m = O\left(\frac{dn^2 \ln \frac{nd}{\delta\varepsilon}}{\varepsilon^4}\right)$ points from q.

2. (lines 6 - 10) For each training point, x_i, determine the largest radius r_i for which

$$\frac{|B(x_i, r_i) \cap T|}{m} \geq \lambda Est(x_i, r_i, S),$$
$$Est(x_i, r_i, S) \leq \gamma.$$

3. (lines 12 - 13) Draw a fresh sample of points from $U \sim q^{O(1/\varepsilon^2)}$, and use it to estimate the probability mass under q of $\cup_{i=1}^{n} B(x_i, r_i)$.

In the first step, we draw a *large* sample from q. While this is considerably larger than the amount of training data we have, we note that samples from q are considered free, and thus do not affect the sample complexity. The reason we need this many samples is simple – unlike p, q is not necessarily regular, and consequently we need enough points to properly estimate q around every training point in S.

The core technical details of *Data_Copy_Detect* are contained within step 2, in which data-copying regions surrounding each training point, x_i, are found. We use $Est(x, r, S)$ and $\frac{|B(x,r)\cap T|}{m}$ as proxies for p and q in Definition 5.2.1, and then search for the maximal radius r_i over which the desired criteria of data-copying are met for these proxies.

The only difficulty in doing this is that this could potentially require checking an infinite number of radii, r_i. Fortunately, this turns out not to be needed because of the following observation – we only need to check radii at which a new point from T is included in the estimation $q_i(r)$. This is because these our estimation for $q_i(r)$ does not change between them meaning that our estimate of the ratio between q and p is maximal nearby these points.

Once we have computed r_i, all that is left is to estimate the data-copy rate by sampling q once more to find the total mass of data-copying region, $\cup_{i=1}^{n} B(x_i, r_i)$.

5.5.1 Performance of Algorithm 6

We now show that given enough data, *Data_Copy_Detect* provides a close approximation of cr_q.

Theorem 5.5.1. *Data_Copy_Detect is a data-copying detector (Definition 5.3.2) with sample complexity at most*

$$m_p(\varepsilon, \delta) = O\left(\frac{d \ln \frac{d}{\delta \varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}\right),$$

for all regular distributions, p.

Theorem 6 shows that our algorithm's sample complexity has standard relationships with the tolerance parameters, ε and δ, along with the input space dimension d. However, it includes an additional factor of $\frac{1}{p_\varepsilon}$, which is a distribution specific factor measuring the regularity of the probability distribution. Thus, our bound cannot be used to give a bound on the amount of data needed without having a bound on p_ε.

We consequently view our upper bound as more akin to a convergence result, as it implies that our algorithm is guaranteed to converge as the amount of data goes towards infinity.

5.5.2 Applying Algorithm 6 to Halfmoons

We now return to the example presented in Figure E.1 and empirically investigate the following question: is our algorithm able to outperform the one given in [3] over this example?

To investigate this, we test both algorithms over a series of distributions by varying the parameter ρ, which is the proportion of points that are "copied." Figure E.1 demonstrates a case in which $\rho = 0.4$. Additionally, we include a parameter, c, for [3]'s algorithm which represents the number of clusters the data is partitioned into (with c-means clustering) prior to running their test. Intuitively, a larger number of clusters means a better chance of detecting more localized data-copying.

The results are summarized in the following table where we indicate whether the algorithm determined a statistically significant amount of data-copying over the given generated

distribution and corresponding training dataset. Full experimental details can be found in Sections E.1 and E.1.3 of the appendix.

Table 5.1. Statistical Significance of data-copying Rates over Halfmoons

Algo	$\mathbf{q = p}$	$\rho = \mathbf{0.1}$	**0.2**	**0.3**	**0.4**
Ours	no	yes	yes	yes	yes
$\mathbf{c = 1}$	no	no	no	no	no
$\mathbf{c = 5}$	no	no	no	no	yes
$\mathbf{c = 10}$	no	no	no	no	yes
$\mathbf{c = 20}$	no	no	no	yes	yes

As the table indicates, our algorithm is able to detect statistically significant data-copying rates in all cases it exists. By contrast, [3]'s test is only capable of doing so when there is a large data-copy rate and when the number of clusters, c, is quite large.

5.6 Is smoothness necessary for data copying detection?

Algorithm 6's performance guarantee requires that the input distribution, p, be regular (Definition 5.4.1). This condition is essential for the algorithm to successfully estimate the probability mass of arbitrarily small balls. Additionally, the parameter, p_ε, plays a key role as it serves as a measure of how "smooth" p is with larger values implying a higher degree of smoothness.

This motivates a natural question – can data copying detection be done over unsmooth data distributions? Unfortunately, the answer turns out to be no. In the following result, we show that if the parameter, p_ε is allowed to be arbitrarily small, then this implies that for any data-copy detector, there exists p for which the sample complexity is arbitrarily large.

Theorem 5.6.1. *Let B be a data-copying detector. Let $\varepsilon = \delta = \frac{1}{3}$. Then, for all integers $\kappa > 0$,*

there exists a probability distribution p such that $\frac{1}{9\kappa} \leq p_\varepsilon \leq \frac{1}{\kappa}$, and $m_p(\varepsilon, \delta) \geq \kappa$, implying that

$$m_p(\varepsilon, \delta) \geq \Omega\left(\frac{1}{p_\varepsilon}\right).$$

Although Theorem 5.6.1 is restricted to regular distributions, it nevertheless demonstrates that a bound on smoothness is essential for data copying detection. In particular, non-regular distributions (with no bound on smoothness) can be thought of as a degenerate case in which $p_\varepsilon = 0$.

Additionally, Theorem 5.6.1 provides a lower bound that complements the Algorithm 6's performance guarantee (Theorem 5.5.1). Both bounds have the same dependence on p_ε implying that our algorithm is optimal at least in regards to p_ε. However, our upper bound is significantly larger in its dependence on d, the ambient dimension, and ε, the tolerance parameter itself.

While closing this gap remains an interesting direction for future work, we note that the existence of a gap isn't too surprising for our algorithm, *Data_Copy_Detect*. This is because *Data_Copy_Detect* essentially relies on manually finding the entire region in which data-copying occurs, and doing this requires precise estimates of p at all points in the training sample.

Conversely, detecting data-copying only requires an *overall* estimate for the data-copying rate, and doesn't necessarily require finding all of the corresponding regions. It is plausible that more sophisticated techniques might able to estimate the data-copy rate *without* directly finding these regions.

5.7 Conclusion

In conclusion, we provide a new modified definition of "data-copying" or generating memorized training samples for generative models that addresses some of the failure modes of previous definitions [3]. We provide an algorithm for detecting data-copying according to our definition, establish performance guarantees, and show that at least some smoothness conditions are needed on the data distribution for successful detection.

With regards to future work, one important direction is in addressing the limitations discussed in section 5.2.2. Our definition and algorithm are centered around the assumption that the goal of a generative model is to output q that is close to p in a mathematical sense. As a result, we are unable to handle cases where the generator tries to generate *transformed* examples that lie outside the support of the training distribution. For example, a generator restricted to outputting black and white images (when trained on color images) would remain completely undetected by our algorithm regardless of the degree with which it copies its training data. To this end, we are very interested in finding generalizations of our framework that are able to capture such broader forms of data-copying.

Appendix A

Appendix for Chapter 1

A.1 Proofs for r-separated distributions

For any distribution $\mathscr{D}$ over $\mathscr{X} \times Y$, it will be convenient to use the following notation: for any measurable $S \subset \mathscr{X}$, let $\mathbb{P}_{\mathscr{D}}[S] = \mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in S]$. The following definition will be central to our proofs.

Definition A.1.1. *Let $\mathscr{D}$ be a distribution over $\mathscr{X} \times Y$. An $(\varepsilon, \gamma, \alpha)$-**decomposition** of $\mathscr{D}$ is a finite set of closed balls $B_1, B_2, \ldots, B_s \subset \mathscr{X}$ each with radius γ such that*

$$\mathbb{P}_{\mathscr{D}}[\cup_1^s B_i] > 1 - \varepsilon,$$

and such that $\mathbb{P}_{\mathscr{D}}[B_i] \geq \alpha > 0$ for $1 \leq i \leq s$.

Lemma A.1.2. *Let $\mathscr{X}$ be a totally bounded metric space. For any distribution $\mathscr{D}$, and $\varepsilon, \gamma > 0$, there exists $\alpha > 0$ such that $\mathscr{D}$ admits a $(\varepsilon, \gamma, \alpha)$-decomposition.*

Proof. Fix any $x \in \mathscr{X}$ and $\varepsilon, \gamma > 0$. Then the sequence of balls $\{S_i = B(x, i)\}$ has union equal to $\mathscr{X}$. Therefore, there exists j such that $P_{\mathscr{D}}(S_j) > 1 - \varepsilon$. Since S_j is totally bounded and complete, it is compact. Let $B^o(x, a)$ denote the open ball centered at x with radius a. Therefore, taking an open cover of S_j, $\{B^o(x, \gamma) : x \in S_j\}$, we can take a finite subcover $\{B_1^o, B_2,^o, \ldots, B_t^o\}$ that cover S_j. Discarding balls such that $\mathbb{P}_{\mathscr{D}}(B_i^o) = 0$ and taking the closure of each ball gives the desired result, with $\alpha = \min_i P_{\mathscr{D}}(B_i)$. □

To prove Theorem 1.3.1, we use the following lemma.

Lemma A.1.3. *Let $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$, and let $B_1, B_2, \ldots, B_s$ be a $(\varepsilon, \gamma, \alpha)$-decomposition of $\mathscr{D}$, and let $r > 3\gamma$. If W is a weight function satisfying the conditions of Theorem 1.3.1, then for any $\delta > 0$ there exists N such that for $n \geq N$, with probability $1 - \delta$ over $S \sim \mathscr{D}^n$, and $w_1, w_2, \ldots, w_n$ learned by W from S,*

$$\sup_{\{x : d(x, \cup_1^s B_i) \leq r - 3\gamma\}} \sum_1^n w_i(x) I_{d(x_i, x) > r} < \frac{1}{3}.$$

Proof. Fix $\delta > 0$, and let Y be the indicator variable defined as

$$Y = \begin{cases} 1 & \text{if } \sup_{\{x : d(x, \cup_1^s B_i) \leq r - 3\gamma\}} \sum_1^n w_i(x) I_{d(x_i, x) > r} \geq \frac{1}{3} \\ 0 & \text{if } \sup_{\{x : d(x, \cup_1^s B_i) \leq r - 3\gamma\}} \sum_1^n w_i(x) I_{d(x_i, x) > r} < \frac{1}{3} \end{cases}.$$

It suffices to show that there exists N such that for all $n \geq N$, $E_{S \sim \mathscr{D}}[Y] \leq \delta$.

Fix $S \sim \mathscr{D}^n$ and suppose that $Y = 1$. Then there exists x^*, B_i^* such that $d(x^*, B_i^*) \leq r - 3\gamma$ and such that

$$\sum_1^n w_i(x^*) I_{d(x_i, x^*) > r} \geq \frac{1}{3}.$$

By definition, B_i has radius γ, so by the triangle inequality, for any $x \in B_i^*$, $d(x, x^*) \leq 2\gamma + r - 3\gamma = r - \gamma$. This implies $x^* \in B(x, r - \gamma)$. Therefore, for any $x \in B_i^*$,

$$\sup_{x' \in B(x, r-\gamma)} \sum_1^n w_i(x') I_{d(x', x_i) > r} \geq \sum_1^n w_i(x^*) I_{d(x^*, x_i) > r} \geq \frac{1}{3}.$$

By the definition of an $(\varepsilon, \gamma, \alpha)$-decomposition, we have that $P_{\mathscr{D}}(B_i^*) \geq \alpha$. As a consequence, we have that

$$\mathbb{E}_{X \sim \mathscr{D}_{\mathscr{X}}} \Big[\sup_{x' \in B(X, r-\gamma)} \sum_1^n w_i(x') I_{||x_i - x'|| > r} \Big] \geq P_{\mathscr{D}}[B_i^*] \frac{1}{3} \geq \frac{\alpha}{3}.$$

Since the previous inequality is guaranteed to hold if $Y = 1$, taking the expectation over S yields

that

$$\mathbb{E}_{S\sim\mathscr{D}^n}\mathbb{E}_{X\sim\mathscr{D}_{\mathscr{X}}}\Big[\sup_{x'\in B(X,r-\gamma)}\sum_1^n w_i(x')I_{||x_i-x'||>r}\Big]\geq\frac{\alpha E[Y]}{3}.$$

By the conditions of Theorem 1.3.1, the left side of the equation must tend to 0 as $n\to\infty$. This implies that the same must hold for the right side. Therefore, $E[Y]$ tends to 0 as $n\to\infty$, and we can select N such that $E[Y]<\delta$ for $n\geq n$, which completes the proof. □

Proof. (**Theorem 1.3.1**) Let W be a weight function that satisfies the condition of Theorem 1.3.1. Fix $\varepsilon,\delta>0$, and $\gamma<r/3$. Applying Lemma A.1.2, let $B_1,B_2,\ldots,B_s$ be an $(\varepsilon,\gamma,\alpha)$-decomposition of $\mathscr{D}$. Let T^+ and T^- be subsets of $\mathscr{X}$ corresponding to the definition of r-separation for $\mathscr{D}$.

For $S\sim\mathscr{D}^n$, let A denote the event that

$$\sup_{\{x:d(x,\cup_1^s B_i)\leq r-3\gamma\}}\sum_1^n w_i(x)I_{d(x_i,x)>r}<\frac{1}{3}.$$

Suppose A holds. Pick a B_i. Since T^+ and T^- have distance greater than $2r$, and $diam(B_i)\leq 2\gamma<r$, either $B_i\cap T^+=\emptyset$ or $B_i\cap T^-=\emptyset$. Note that for n sufficiently large, both cannot be empty since $P_{\mathscr{D}}(B_i)\geq\alpha>0$ and each x in the support of $\mathscr{D}$ is either in T^+ or T^-.

Without loss of generality, $B_i\cap T^-=\emptyset$. Then $B_i\cap T^+\neq\emptyset$. B_i has diameter 2γ. Thus $d(B_i,T^-)>2r-2\gamma$. Let $x\in B(B_i,r-3\gamma)$. Then if $(x_j,-)\in S$, by the triangle inequality, $d(x,x_j)>2r-2\gamma-(r-3\gamma)=r+\gamma$.

Substituting this and using event A, we have that

$$\sum_1^n w_i^S(x)I_{(x_i,-)\in S}\leq\sum_1^n w_i^S(x)I_{d(x_i,x)>r}<\frac{1}{3}.$$

It follows that $W_S(x)=+1$. An analogous argument holds for $B_i\cap T^+=\emptyset$. This implies that W_S is astute with radius $r-3\gamma$ over all B_i.

$\cup B_i$ has measure at least $1-\varepsilon$. By Lemma A.1.3, for any $\delta>0$ event A holds with

probability $1-\delta$ for n sufficiently large. Therefore, for n sufficiently large, we see that $A_{r-3\gamma}(W_S, \mathscr{D}) \geq 1-\varepsilon$ with probabiltiy $1-\delta$. Because ε, δ and γ were arbitrary, it follows that W is r-consistent, as desired.

□

Proof. (**Corollary 1.3.3**) For any $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \subset \mathscr{X} \times \{\pm 1\}$, let $w_i^S(x)$ be 1 if and only if x_i is one of the k_n nearest neighbors of x in the set $S_{\mathscr{X}} = \{x_1, x_2, \ldots x_n\}$. Let $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$. By Theorem 1.3.1, it suffices to show that for any $0 < a < b$,

$$\lim_{n\to\infty} \mathbb{E}_{X\sim\mathscr{D}_{\mathscr{X}}}[\mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x'\in B(x,a)} \sum_1^n w_i^S(x') I_{d(x_i,x')>b}]] = 0.$$

Fix $0 < a < b$, and let $\varepsilon > 0$.

Pick $\gamma > 0$ such that $a + 2\gamma < b$. This is possible for any $a < b$. Let $B_1, B_2, \ldots, B_s$ be an $(\varepsilon, \gamma, \alpha)$-decomposition of $\mathscr{D}$. By applying a Chernoff bound followed by a union bound, for any $\delta > 0$ there exists n such that with probability $1-\delta$ over $S \sim \mathscr{D}^n$, each B_i satisfies $|B_i \cap S_{\mathscr{X}}| \geq \frac{n\alpha}{2}$. Furthermore, if n is sufficiently large, then $\frac{n\alpha}{2} > k_n$ holds as well.

Consider any $x \in B_i$, and$x' \in B(x,a)$. B_i has radius γ and also satisfies $|B_i \cap S_{\mathscr{X}}| > k_n$. Therefore, there are at least k_n points within distance $a + 2\gamma$ of x. Because $a + 2\gamma < b$, it follows that none of the k_n nearest neighbors of x' can have distance more than b from x'. In particular,

$$\sum_1^n w_i^S(x') I_{d(x_i,x')>b} = 0.$$

Since B_i, x and x' were arbitrary, we have that for all $x \in \cup B_i$,

$$\sup_{x'\in B(x,a)} \sum_1^n w_i^S(x') I_{d(x_i,x')>b} \leq \begin{cases} 0 & |B_i \cap S_{\mathscr{X}}| \geq \frac{n\alpha}{2}, 1 \leq i \leq s \\ 1 & \text{otherwise} \end{cases}$$

Since $X \in \cup_1^s B_i$ with probability at least $1-\varepsilon$, and since $|B_i \cap S_{\mathscr{X}}| \geq \frac{n\alpha}{2}, 1 \leq i \leq s$ with

probability at least $1-\delta$, it follows that

$$\mathbb{E}_{X\sim\mathscr{D}}[\mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x'\in B(x,a)}\sum_1^n w_i^S(x')I_{d(x_i,x')>b}]] \leq (1-\delta-\varepsilon)0+\delta+\varepsilon = \delta+\varepsilon,$$

which can be made arbitrarily small as ε and δ were arbitrary. Therefore, the limit as n approaches infinity is 0, as desired. □

Proof. (**Corollary 1.3.4**) Let $\mathscr{D}$ be a distribution over $\mathscr{X}\times\{\pm 1\}$. By Theorem 1.3.1, it suffices to show that for any $0<a<b$,

$$\lim_{n\to\infty}\mathbb{E}_{X\sim\mathscr{D}}[\mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x'\in B(x,a)}\sum_1^n w_i^S(x')I_{d(x_i,x')>b}]] = 0.$$

Fix $0<a<b$, and let $\varepsilon>0$.

Pick $\gamma>0$ be such that $a+2\gamma<b$. Let $B_1,B_2,\ldots,B_s$ be an $(\varepsilon,\gamma,\alpha)$-decomposition of $\mathscr{D}$. By applying a Chernoff bound, for any $\delta>0$ there exists n such that with probability $1-\delta$ over $S\sim\mathscr{D}^n$, each B_i satisfies $|B_i\cap S_{\mathscr{X}}|\geq\frac{n\alpha}{2}$.

Next, consider any $x_i,x_j\in S_{\mathscr{X}}$, and let x be a point such that $d(x_i,x)\leq a+2\gamma$ and $d(x_j,x)>b$. Then we have that

$$\frac{w_j^S(x)}{w_i^S(x)} = \frac{K(\frac{d(x_j,x)}{h_n})}{K(\frac{d(x_i,x)}{h_n})}.$$

Because $b>a+2\gamma$, $\frac{d(x_j,x)}{d(x_i,x)}>1$. Therefore, since $\lim_{n\to\infty}h_n=0$ and $\lim_{x\to\infty}\frac{K(cx)}{K(x)}=0$ for $c>1$, it follows that for any $\beta>0$, there exists N such that for $n\geq N$,

$$\frac{w_j^S(x)}{w_i^S(x)}\leq\frac{\alpha\beta}{2}.$$

Fix any such β, and consider any x with $d(x,B_i)\leq a$. Then $d(x,x')\leq a+2\gamma<b$ for any $x'\in B_i$. Recall that B_i contains at least $\frac{n\alpha}{2}$ points, and let $c=\min_{i,d(x_i,x)\leq a+2\gamma}w_i(x)$. Then it

follows that

$$\begin{aligned}\sum_1^n w_i^S(x) I_{d(x_i,x)>b} &\overset{(a)}{=} \frac{\sum_1^n w_i^S(x) I_{d(x_i,x)>b}}{\sum_1^n w_i^S(x)} \\ &\overset{(b)}{\leq} \frac{\sum_1^n w_i^S(x) I_{d(x_i,x)>b}}{\sum_1^n w_i^S(x) I_{d(x_i,x)\leq a+2\gamma}} \\ &\overset{(c)}{\leq} \frac{nc\frac{\alpha\beta}{2}}{\frac{n\alpha}{2}c} \\ &= \beta\end{aligned}$$

(a) holds because the weights always sum to 1. (b) holds because we are reducing the denominator. (c) holds because there are at least $\frac{n\alpha}{2}$ points in B_i, with c being the minimum weight (stated above). The numerator is a result of the inequality shown above in which $w_j^S(x)/w_i^S(x) \leq \alpha\beta/2$ if $d(x_j,x) > b$ and $d(x_i,x) \leq a+2\gamma$.

Using this, we get the following bound:

$$\sup_{x' \in B(X,a)} \sum_1^n w_i^S(x') I_{d(x_i,x')>b} \leq \begin{cases} \beta & x \in \cup_1^s B_i, |B_i \cap S_{\mathscr{X}}| \geq \frac{n\alpha}{2}, 1 \leq i \leq s \\ 1 & \text{otherwise} \end{cases}$$

Since $x \in \cup_1^s B_i$ with probability $1-\varepsilon$, and since $|B_i \cap S_{\mathscr{X}}| \geq \frac{n\alpha}{2}, 1 \leq i \leq s$ with probability $1-\delta$, it follows that

$$\mathbb{E}_{X\sim\mathscr{D}}[\mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x' \in B(x,a)} \sum_1^n w_i^S(x') I_{d(x_i,x')>b}]] \leq (1-\delta-\varepsilon)\beta + \delta + \varepsilon.$$

which can be made arbitrarily small as ε, β, and δ were arbitrary. Therefore, the limit as n approaches infinity is 0, as desired. □

A.2 Proofs for general distributions

Lemma A.2.1. *Let $B_1, \ldots, B_s$ be a $(\varepsilon, \alpha, \gamma)$ decomposition of $\mathscr{D}$ over $\mathscr{X} \times \{\pm 1\}$. Let $U \subseteq [s]$. Then if $n \geq O(\frac{s2^{2s}\log(1/\delta)}{\varepsilon^2})$, then with probability at least $1-\delta$, for all U we have:*

$$|\mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in U} B_i, y = +] - \mathbb{P}_{(x,y)\sim\mathscr{D}_S}[x \in \cup_{i\in U} B_i, y = +]| \leq \varepsilon,$$

$$|\mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in U} B_i, y = -] - \mathbb{P}_{(x,y)\sim\mathscr{D}_S}[x \in \cup_{i\in U} B_i, y = -]| \leq \varepsilon.$$

Proof. For any given $U \subseteq [s]$, by a Chernoff bound we have that

$$|\mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in U} B_i, y = +] - \mathbb{P}_{(x,y)\sim\mathscr{D}_S}[x \in \cup_{i\in U} B_i, y = +]| > \varepsilon$$

with probability at most $\frac{\delta}{2^{s+1}}$. Taking a union bound over all U, we see that with probability $1-\frac{\delta}{2}$,

$$|\mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in U} B_i, y = +] - \mathbb{P}_{(x,y)\sim\mathscr{D}_S}[x \in \cup_{i\in U} B_i, y = +]| \leq \varepsilon$$

for all $U \subseteq [m]$. Applying the same to $y = -1$ and taking a union bound implies the result. □

Lemma A.2.2. *Let M be a classification algorithm over $\mathscr{X} \times \{\pm 1\}$, $r > 0$ be a radius, and $\mathscr{D}$ be a distribution over $\mathscr{X} \times \{\pm 1\}$. Then for any ε, δ over $(0,1)$, and for all γ over $(0, r/2)$, there exists N such that for $n \geq N$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$,*

$$A_{r-\gamma}(M_S, \mathscr{D}) \geq A_r(M_S, \mathscr{D}_S) - \varepsilon,$$

where $\mathscr{D}_S$ denotes the uniform distribution over S.

Proof. (**Lemma A.2.2**) Fix $\varepsilon, \delta > 0$ and $\gamma < r/2$. Applying Lemma A.1.2, let $B_1, \ldots, B_s$ be a $(\varepsilon, \alpha, \gamma)$ decomposition of $\mathscr{D}$.

Let T be the subset of S such that M_S is astute at T with radius r. Define:

$$I_T^+ = \{i | (x_j, +) \in T, x_j \in B_i\}$$

$$I_T^- = \{i | (x_j, -) \in T, x_j \in B_i\}.$$

Observe that $I_T^+ \cap I_T^- = \emptyset$. To see this, notice that B_i has radius $\gamma < r/2$. This implies that any $(x_j, +), (x_k, -) \in B_i$ would force M_S to not be astute at either of those points. Thus we an think of I_T^+ being the set of positively labeled balls, and I_T^- being the set of negatively labeled balls.

Let $B^+ = \cup_{i \in I_T^+} B_i$ and $B^- = \cup_{i \in I_T^-} B_i$. Our strategy will be to argue that M_S must be robust with radius $r - 2\gamma$ at $B^+ \cup B^-$, and then to observe that $\mathbb{P}_{\mathscr{D}}[(B^+, +)] + \mathbb{P}_{\mathscr{D}}[(B^-, -)]$ must be close to $A_r(M_S, \mathscr{D}_S)$.

Let $T_{\mathscr{X}} \subset \mathscr{X}$ denote the set of all x_i such that $(x_i, y_i) \in T$. By the definitions of $\mathscr{D}_S$ and T, we have that

$$\begin{aligned} A_r(M_S, \mathscr{D}_S) &= \frac{|T|}{n} \\ &= \frac{|T_{\mathscr{X}} \cap B^+|}{n} + \frac{|T_{\mathscr{X}} \cap B^-|}{n} + \frac{|T_{\mathscr{X}} \setminus (B^+ \cup B^-)|}{n}. \end{aligned}$$

If $x_i \in \cup_1^s B_j$ and $x_i \in T_{\mathscr{X}}$, then by definition, $x \in (B^+ \cup B^-)$. Therefore, $T_{\mathscr{X}} \setminus (B^+ \cup B^-)$ consists of $x_i \notin \cup_1^s B_j$. Using this, we see that

$$\begin{aligned} A_r(M_S, \mathscr{D}_S) &= \frac{|T_{\mathscr{X}} \cap B^+|}{n} + \frac{|T_{\mathscr{X}} \cap B^-|}{n} + \frac{|T_{\mathscr{X}} \setminus (B^+ \cup B^-)|}{n} \\ &\leq \mathbb{P}_{(x,y) \sim \mathscr{D}_S}[x \in B^+, y = +] + \mathbb{P}_{(x,y) \sim \mathscr{D}_S}[x \in B^-, y = -] + \mathbb{P}_{(x,y) \sim \mathscr{D}_S}[x \notin \cup_1^s B_j]. \end{aligned}$$

If n is sufficiently large, then by Lemma A.2.1, each term on the right is within ε of its corresponding probability over $\mathscr{D}$. Thus we see that with probability $1 - \delta$,

$$A_r(M_S, \mathscr{D}_S) \leq \mathbb{P}_{(x,y) \sim \mathscr{D}}[x \in \cup_{i \in I_T^+} B_i, y = +] + \mathbb{P}_{(x,y) \sim \mathscr{D}}[x \in \cup_{i \in I_T^-}, y = -] + 4\varepsilon. \tag{A.1}$$

Observe that if M_S is robust with radius r at $x_j \in B_i$, then it is robust with radius $r-2\gamma$ at all $x \in B_i$. Furthermore, for $x_j \in \cup_{i\in I_T^+} B_i$, M_S is astute at $(x_j,+1)$ with radius r. Therefore $M_S(x) = +1$ for all $x \in \cup_{i\in I_T^+} B_i$. Consequently,

$$\begin{aligned} A_{r-2\gamma}(M_S,\mathscr{D}) &\geq \mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in I_T^+} B_i, y = +] + \mathbb{P}_{(x,y)\sim\mathscr{D}}[x \in \cup_{i\in I_T^-} B_i, y = -] \\ &\geq A_r(M_S,\mathscr{D}_S) - 4\varepsilon \text{ (by equation } A.1). \end{aligned}$$

Since this equation holds with probability $1-\delta$, and since ε and γ were arbitrary, the result follows. □

Proof. (**Theorem 1.4.3**) For convenience, we let W' represent the weight function described by $RobustNonPar(S,W,r)$. In particular, W'_S and W_{S_r} are the same classifier, where S_r denotes the largest r-separated subset of S.

Fix $\varepsilon,\delta > 0$, and let $0 < \gamma < r$. For convenience, let

$$Z_i = \sup_{x\in B(x_i,r-\gamma)} \sum_{j=1}^{m} w_j^{S_r}(x) I_{||x_j-x||>r}.$$

Because W fulfills the conditions of Theorem 1.4.3, there exists N such that for $n > N$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$, $\frac{1}{m}\sum_{i=1}^m Z_i < \varepsilon$. Therefore, there exist at most $3m\varepsilon$ values of i for which $Z_i > \frac{1}{3}$.

Since S_r is r-separated, it follows that

$$\sup_{x\in B(x_i,r-\gamma)} \sum_{1}^{m} w_j^{S_r}(x) I_{y_j\neq y_i} \leq Z_i.$$

Consequently, if $Z_i \leq \frac{1}{3}$, then $W_{S_r}(x) = y_i$ for all $x \in B(x_i, r-\gamma)$. Let $\mathscr{D}_S$ denote the uniform distribution over S. Then we have that

$$A_{r-\gamma}(W'_S,\mathscr{D}_S) = A_{r-\gamma}(W_{S_r},\mathscr{D}_S) \geq \frac{|S_r|}{n} - 3\varepsilon.$$

Observe that for n sufficiently large, with probability $1-\delta$, $|A_r(g_r^*, \mathscr{D}) - A_r(g_r^*, \mathscr{D}_S)| \leq \varepsilon$. The maximum possible astuteness over $\mathscr{D}_S$ is $\frac{|S_r|}{n}$ since no classifier can be astute at 2 oppositely labeled points with distance at most $2r$. Therefore, with probability $1-2\delta$,

$$A_{r-\gamma}(W_S', \mathscr{D}_S) \geq A_r(g_r^*, \mathscr{D}) - 4\varepsilon.$$

By Lemma A.2.2, for n sufficiently large, with probability $1-\delta$

$$A_{r-2\gamma}(W_S', \mathscr{D}) \geq A_{r-\gamma}(W_S', \mathscr{D}_S) - \varepsilon.$$

Therefore, for n sufficiently large, with probability $1-3\delta$ over $S \sim \mathscr{D}$,

$$A_{r-2\gamma}(W_S', \mathscr{D}) \geq A_r(g_r^*, \mathscr{D}) - 5\varepsilon.$$

Since ε, δ, and γ were arbitrary, we are done. □

The following two quick lemmas are used for the proofs of Corollaries 1.4.4 and 1.4.5.

Lemma A.2.3. *Let $B_1, B_2, \ldots, B_s \subset \mathscr{X}$ denote s balls. Let $T \subset \mathscr{X}$ satisfy $|T \cap \cup_1^s B_i| = m$. Let*

$$I_k \subseteq [s] = \{i : |B_i \cap T| \geq k\}.$$

Then $|\cup_{i \in I_k} B_i \cap T| \geq m - ks$.

Proof. For any $j \notin I_k$, $|B_j \cap T| < k$. Since there are at most s such j, it follows that $|\cup_{i \notin I_k} B_i \cap T| < ks$. Taking the complement implies the result. □

Lemma A.2.4. *Let S be a finite subset of $\mathscr{X} \times \{\pm 1\}$. For any $r > 0$, let S_r denote the largest r-separated subset of S. Then $|S_r| \geq \frac{|S|}{2}$.*

Proof. Let $S = \{(x_1, y_1), (x_2, y_2), \ldots (x_n, y_n)\}$. Define:

$$S_+ = \{(x_i, y_i) : y_i = +1\}$$

$$S_- = \{(x_i, y_i) : y_i = -1\}.$$

Observe that S_+ and S_- are both r-separated and have union S. Therefore one must have cardinality at least $\frac{|S|}{2}$, which implies the same about $|S_r|$. □

Proof. (**Corollary 1.4.4**) For convenience, we let W' represent the weight function described by $RobustNonPar(S, W, r)$. In particular, W'_S and W_{S_r} are the same classifier, where S_r denotes the largest r-separated subset of S.

Relabel the points in S so that

$$S_r = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\},$$

with $m \leq n$. We will also let $S_r^{\mathscr{X}} = \{x_1, x_2, \ldots, x_m\}$.

By Theorem 1.4.3, it suffices to show that for any $0 < a < b$,

$$\lim_{n\to\infty} \mathbb{E}_{S\sim\mathscr{D}^n}[\frac{1}{m}\sum_{i=1}^{m} \sup_{x\in B(x_i,a)} \sum_{j=1}^{m} w_j^{S_r}(x) I_{d(x_i,x)>b}] = 0,$$

where w_j denote the weight functions corresponding to W. Fix $0 < a < b$, and let $\varepsilon > 0$.

Pick $\gamma > 0$ be such that $a + 2\gamma < b$. Let $B_1, B_2, \ldots, B_s$ be a $(\varepsilon, \gamma, \alpha)$ decomposition of $\mathscr{D}$. By applying a Chernoff bound, for any $\delta > 0$ there exists n_0 such that for $n \geq n_0$, with probability $1 - \delta$ over $S \sim \mathscr{D}^n$,

$$|S_{\mathscr{X}} \cap \cup_1^s B_i| \geq (1 - 2\varepsilon)n.$$

By Lemma A.2.4, $\frac{m}{n} \geq \frac{1}{2}$. It follows that $|S_r^{\mathscr{X}} \cap \cup_1^s B_i| \geq m(1 - 4\varepsilon)$.

Let

$$J = \{i : |B_i \cap S_r^{\mathscr{X}}| \geq m\frac{\varepsilon}{s}\}.$$

By Lemma A.2.3 it follows that $|S_r^{\mathscr{X}} \cap \cup_{i\in J} B_i| \geq m(1-4\varepsilon) - m\varepsilon = m(1-5\varepsilon)$.

Next, observe that if n is sufficiently large, then

$$\frac{k_n}{m} \leq \frac{2k_n}{n} \leq \frac{\varepsilon}{s}.$$

Therefore, $|B_i \cap S_r^X|_r \geq k_n$ for $i \in J$.

Fix any B_j with $j \in J$, and consider x with $d(x,B_j) \leq a$. Then $d(x,x') \leq a+2\gamma < b$ for any $x' \in B_j$. Therefore, since $|S_r^X \cap B_i| \geq k_n$, all k_n-nearest neighbors of x have distance at most b to x. This implies that

$$\sum_1^m w_i^{S_r}(x) I_{d(x_i,x)>b} = 0.$$

For convenience, let

$$f(x_i) = \sup_{x\in B(x_i,a)} \sum_{j=1}^m w_j^{S_r}(x) I_{d(x,x_j)>b}.$$

For $x_i \in \cup_{j\in J} B_j$, any $x \in B(x_i,a)$ trivially satisfies $d(x,B_i) \leq a$. Therefore, $f(x_i) = 0$ Since $|S_r^{\mathscr{X}} \cap \cup_{j\in J} B_j| \geq m(1-5\varepsilon)$, and $f(x_i) \leq 1$ for all $1 \leq i \leq m$, we have that

$$\begin{aligned}
\frac{1}{m}\sum_1^m f(x_i) &= \frac{1}{m}\Big(\sum_{x_i\in\cup_{i\in J}B_i} f(x_i) + \sum_{x_i\notin\cup_{i\in J}B_i} f(x_i)\Big) \\
&\leq \frac{1}{m}(0 + 5\varepsilon m(1)) \\
&= 5\varepsilon.
\end{aligned}$$

Since all of our equations hold with probability $1-\delta$ over S for sufficiently large n, this last one does as well. Since this entirely expression is always at most 1 (regardless of S), and since δ, ε

were arbitrary, we have that

$$\lim_{n\to\infty} E_{S\sim\mathscr{D}^n}[\frac{1}{m}\sum_{1}^{m} f(x_i)] = 0,$$

which completes the proof. □

Proof. (**Corollary 1.4.5**) For convenience, we let W' represent the weight function described by $RobustNonPar(S,W,r)$. In particular, W'_S and W_{S_r} are the same classifier, where S_r denotes the largest r-separated subset of S.

Relabel the points in S so that

$$S_r = \{(x_1,y_1),(x_2,y_2),\ldots,(x_m,y_m)\},$$

with $m \leq n$. We will also let $S_r^{\mathscr{X}} = \{x_1,x_2,\ldots,x_m\}$.

By Theorem 1.4.3, it suffices to show that for any $0 < a < b$,

$$\lim_{n\to\infty} \mathbb{E}_{S\sim\mathscr{D}^n}[\frac{1}{m}\sum_{i=1}^{m} \sup_{x\in B(x_i,a)} \sum_{j=1}^{m} w_j^{S_r}(x) I_{d(x_i,x)>b}] = 0,$$

where w_j are the weight functions corresponding to W. Fix $0 < a < b$, and let $\varepsilon > 0$.

Pick $\gamma > 0$ be such that $a + 2\gamma < b$. Let $B_1,B_2,\ldots,B_s$ be a $(\varepsilon,\gamma,\alpha)$ decomposition of $\mathscr{D}$. By applying a Chernoff bound, for any $\delta > 0$ there exists n_0 such that for $n \geq n_0$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$,

$$|S_{\mathscr{X}} \cap \cup_1^s B_i| \geq (1-2\varepsilon)n.$$

By Lemma A.2.4, $\frac{m}{n} \geq \frac{1}{2}$. It follows that $|S_r^{\mathscr{X}} \cap \cup_1^s B_i| \geq m(1-4\varepsilon)$.

Let

$$J = \{i : |B_i \cap S_r^{\mathscr{X}}| \geq \frac{m\varepsilon}{s}\}.$$

By Lemma A.2.3, $|S_r^{\mathscr{X}} \cap \cup_{i\in J} B_i| \geq m(1-4\varepsilon) - m\varepsilon = m(1-5\varepsilon)$.

Next, consider any $x_i,x_j \in S_r^{\mathscr{X}}$, and let x be a point such that $d(x_i,x) \leq a+2\gamma$ and $d(x_j,x) > b$. Recall that W is constructed from kernel function K and window parameter h_n. We

then have that

$$\frac{w_j^S(x)}{w_i^S(x)} = \frac{K(\frac{d(x_j,x)}{h_n})}{K(\frac{d(x_i,x)}{h_n})}. \tag{A.2}$$

Because $b > a + 2\gamma$, $\frac{d(x_j,x)}{d(x_i,x)} > 1$. Fix any $\beta > 0$. Because $\lim_{n\to\infty} h_n = 0$ and $\lim_{x\to\infty} \frac{K(cx)}{K(x)} = 0$ for $c > 1$, there exists N such that for $n \geq N$,

$$\frac{w_j^S(x)}{w_i^S(x)} \leq \frac{\beta\varepsilon}{s}.$$

Fix B_j with $j \in J$, and consider x with $d(x, B_j) \leq a$. By the triangle inequality, $d(x,x') \leq a + 2\gamma$ for all $x' \in B_j$. Then we have the following,

$$\begin{aligned}
\sum_1^m w_i^{S_r}(x) I_{d(x_i,x)>b} &\overset{(a)}{=} \frac{\sum_1^m w_i^{S_r}(x) I_{d(x_i,x)>b}}{\sum_1^m w_i^{S_r}(x)} \\
&\overset{(b)}{\leq} \frac{\sum_1^m w_i^{S_r}(x) I_{d(x_i,x)>b}}{\sum_{x_i\in B_j} w_i^{S_r}(x)} \\
&\overset{(c)}{\leq} \frac{m \sup_{x_i: d(x_i,x)>b} w_i^{S_r}(x)}{m\varepsilon/s \inf_{x_i\in B_j} w_i^{S_r}(x)} \\
&\overset{(d)}{\leq} \frac{\beta\varepsilon/s}{\varepsilon/s} = \beta.
\end{aligned} \tag{A.3}$$

Equation (a) holds because the total sum of weights is always 1, (b) because all weights are nonnegative, (c) because $|B_j \cap S_r^{\mathscr{X}}| \geq m\varepsilon/s$, and (d) because of equation A.2.

Let

$$Z_i = \sup_{x\in B(x_i,a)} \sum_{j=1}^m w_j^{S_r}(x) I_{d(x,x_j)>b}.$$

For $x_i \in \cup_1^t B_j$, any $x \in B(x_i, a)$ trivially satisfies $d(x, B_i) \leq a$. By equation A.3, it follows that

$Z_i \leq \beta$. Since $|\cup_{j\in J} B_j \cap S_r^{\mathscr{X}}| \geq m(1-5\varepsilon)$ and $Z_i \leq 1$ for all $1 \leq i \leq m$, we have that

$$\begin{aligned}\frac{1}{m}\sum_1^m Z_i &= \frac{1}{m}\Big(\sum_{x_i\in\cup_{j\in J}B_j} Z_i + \sum_{x_i\notin\cup_{j\in J}B_j} Z_i\Big)\\ &\leq (1-5\varepsilon)\beta + 5\varepsilon.\end{aligned}$$

Since all of our equations hold with probability $1-\delta$ over S for sufficiently large n, this last one does as well. Since this entire expression is always at most 1 (regardless of S), and since $\delta, \varepsilon, \beta$ were arbitrary, we have that

$$\lim_{n\to\infty} E_{S\sim\mathscr{D}^n}[\frac{1}{m}\sum_1^m Z_i] = 0,$$

which completes the proof.

□

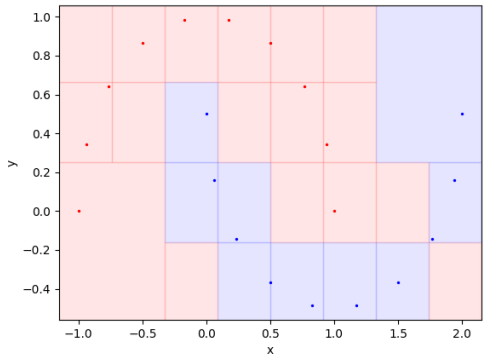
A.3 Experimental Details

A.3.1 Optimal attacks against histogram classifiers

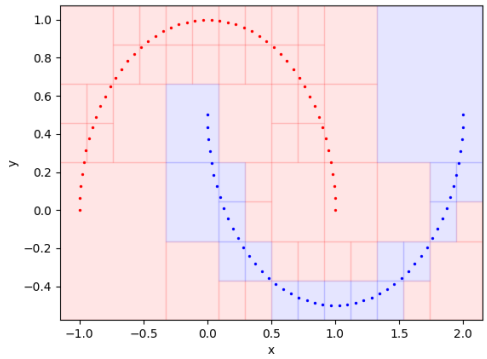
Let H be a histogram classifier, and let (x,y) be any labeled example. Let $r > 0$ be some fixed robustness radius. Recall that an *adversarial example* against H at (x,y) is any x' such that $x' \in B(x,r)$ and $H(x') \neq y$. Note that if $H(x) \neq y$, then x itself is an adversarial example. Conversely, if H is astute at (x,y) with radius r, then no adversarial example exists.

For arbitrary classifiers, finding adversarial examples at a given point can be challenging. However, recent work (Yang et. al. 2019) has shown that for non-parametric classifiers, there are tractable methods for doing so. The key insight is that non-parametric classifiers can be construed as a partitioning of input space into convex cells, with each cell having a given label. For example, Figure A.1 gives a visualization for these cells in a histogram classifier.

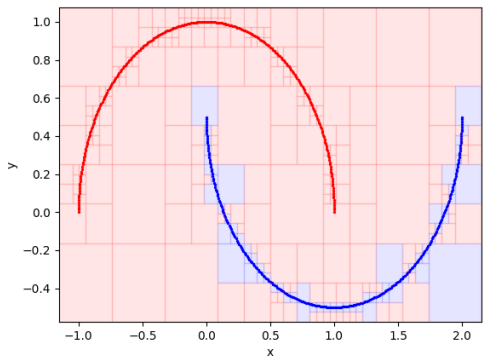
Because these cells are convex, finding an adversarial example for H at (x,y) (here x is a point in $\mathbb{R}^2$, and y is a label) amounts to finding the closest cell $c \in H$ to x such that $H(c) \neq y$.



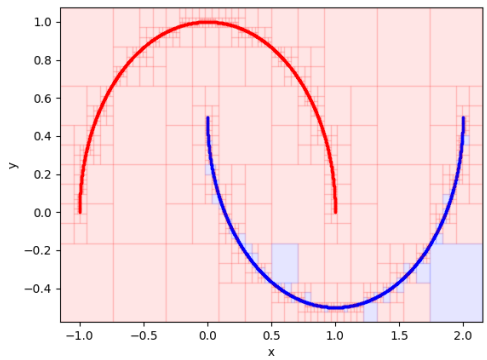
(a) Training Size = 20



(b) Training Size = 50



(c) Training Size = 500



(d) Training Size = 3000

Figure A.1. 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

While Yang et. al. (Yang et. al. 2019) presents convex programming algorithms for doing this, the case of histograms in the ℓ_∞ metric is much simpler.

As stated in definition 10, a histogram partitions the input space into hypercubes by iteratively splitting each cube into 2^d cubes with half the length. Therefore, the cells of a histogram are all hypercubes of varying sizes. For cell c, let $s(c)$ denote the length of the cube that c corresponds to, and let $H(c)$ denote the label H assigns to c. The key observation is that c contains an adversarial example for (x,y) if and only if $d(c,x) \leq s(c)/2 + r$, and $H(c) \neq y$. This yields the following algorithm:

Algorithm 7 was further optimized by utilizing nearest-neighbor type algorithms to find the "closest" cells to x. This was done by grouping cells by their radii, and utilizing a separate nearest-neighbor data structure for all cells of a given radius.

Although this algorithm doesn't have the same performance metrics as those presented in (Yang et. al. 2019), it was easily sufficient for computing the empirical astuteness for our experiments.

Algorithm 7: Optimal attack algorithm for Histogram Classifiers

```
Input: Histogram H, labeled point (x, y) ∈ ℝ² × {±1}, robustness radius r;
for cell c ∈ H do
    if d(c, x) ≤ s(c)/2 + r and H(c) ≠ y then
        Return c
    end if
end for
```

Appendix B

Appendix for Chapter 2

B.1 Further Details of Definitions and Theorems

B.1.1 Non-Parametric Classifiers

In this section, we precisely define weight functions, histogram classifiers and kernel classifiers.

Definition B.1.1. *[12] A **weight function** W is a non-parametric classifier with the following properties.*

1. *Given input $S = \{(x_1, y_1), (x_2, y_2,), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$, W constructs functions $w_1^S, w_2^S, \ldots, w_n^S$: $\mathbb{R}^d \to [0,1]$ such that for all $x \in \mathbb{R}^d$, $\sum_1^n w_i^S(x) = 1$. The functions w_i^S are allowed to depend on $x_1, x_2, \ldots x_n$ but must be independent of $y_1, y_2, \ldots, y_n$.*

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.

Definition B.1.2. *A **histogram classifier**, H, is a non-parametric classification algorithm over $\mathbb{R}^d \times \{\pm 1\}$ that works as follows. For a distribution $\mathscr{D}$ over $\mathbb{R} \times \{\pm 1\}$, H takes $S = \{(x_i, y_i) : 1 \leq$*

$i \leq n\} \sim \mathscr{D}^n$ as input. Let k_i be a sequence with $\lim_{i\to\infty} k_i = \infty$ and $\lim_{i\to\infty} \frac{k_i}{i} = 0$. H constructs a set of hypercubes $C = \{c_1, c_2, \ldots, c_m\}$ 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,

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

Definition B.1.3. *A **partitioning rule** is a weight function W over $\mathscr{X} \times \{\pm 1\}$ constructed in the following manner. Given $S = \{(x_i, y_i)\} \sim \mathscr{D}^n$, as a function of $\{x_1, \ldots, x_n\}$, we partition $\mathbb{R}^d$ into regions with $A(x)$ denoting the region containing x. Then, for any $x \in \mathbb{R}^d$ we have*

$$w_i^S(x) = \begin{cases} 1 & x_i \in A(x) \\ 0 & otherwise \end{cases}.$$

To achieve $\sum w_i^S(x) = 1$, we can simply normalize weights for any x by $\sum_1^n w_i^S(X)$.

Definition B.1.4. *A **kernel classifier** is a weight function W over $\mathbb{R}^d \times \{\pm 1\}$ constructed from function $K : \mathbb{R}^+ \cup \{0\} \to \mathbb{R}^+$ and some sequence $\{h_n\} \subset \mathbb{R}^+$ in the following manner. Given $S = \{(x_i, y_i)\} \sim \mathscr{D}^n$, we have*

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

Then, as above, W has output

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

B.1.2 Splitting Numbers

We refer to definitions 2.4.2 and 2.4.3.

The main idea behind splitting numbers is that they allow us to ensure uniform convergence properties 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 clarity, we include a quick example in which we bound the splitting number for a given weight function.

Example:

Let W denote any kernel classifier corresponding such that $K : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ is a decreasing function. For any $S \sim \mathscr{D}^n$, observe that the condition $w_i^S(x) \geq \beta$ precisely corresponds to $\rho(x, x_i) \leq \gamma$ for some value of γ. This is because $w_i^S(x) > w_j^S(x)$ if and only if $\rho(x, x_i) < \rho(x, x_j)$. Thus, the regions $W_{x,\alpha,\beta}$ correspond to $\{i : \rho(x, x_i) \leq \gamma\}$, where γ is a positive real number that depends on x, α, β. These sets precisely correspond to subsets of S that are contained within $B(x, \gamma)$. Since balls have VC dimension at most $d+2$, by Sauer's lemma, the number of subsets of S that can be obtained in this manner is $O(n^{d+2})$. Therefore, we have that $T(W, S) = O(n^{d+2})$ for all $S \sim \mathscr{D}^n$.

B.1.3 Stone's Theorem

Theorem B.1.5. *[13] Let W be weight function over $\mathbb{R}^d \times \{\pm 1\}$. Suppose the following conditions hold for any distribution $\mathscr{D}$ over $\mathbb{R}^d \times \{\pm 1\}$. Let X be a random variable with distribution*

$\mathscr{D}_{\mathbb{R}^d}$, *and* $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$. *All expectations are taken over* X *and* S.

1. There is a constant c *such that, for every nonnegative measurable function* f *satisfying* $\mathbb{E}[f(X)] < \infty$, *and* $\mathbb{E}[\sum_1^n w_i^S(X) f(x_i)] \leq c\mathbb{E}[f(x)]$.

2. $\forall a > 0$, $\lim_{n\to\infty} \mathbb{E}[\sum_1^n w_i^S(x) I_{||x_i - X|| > a||}] = 0$.

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

Then W *is consistent.*

B.2 Proofs

Notation:

- We let ρ denote our distance metric over $\mathbb{R}^d$. For sets $X_1, X_2 \subset \mathbb{R}^d$, we let $\rho(X_1, X_2) = \inf_{x_1 \in X_1, x_2 \in X_2} \rho(x_1, x_2)$.
- For any $x \in \mathbb{R}^d$, $B(x,a) = \{x : \rho(x,x') \leq a\}$.
- For any measure over $\mathbb{R}^d$, μ, we let $supp(\mu) = \{x : \mu(B(x,a)) > 0 \text{ for all } a > 0\}$.
- Given some measure μ over $\mathbb{R}^d$ and some $x \in \mathbb{R}^d$, we let $r_p(x)$ denote the probability radius (Definition 2.4.2) of x with probability p. that is, $r_p(x) = \inf\{r : \mu(B(x,r)) \geq p\}$.
- For weight function W and training sample S, we let W_S denote the weight function learned by W from S.

B.2.1 Proofs of Theorems 2.3.3 and 2.3.4

Proof. (Theorem 2.3.3) Let $\mathscr{D} = (\mu, \eta)$ be a data distribution, and let μ^+, μ^- be as described in Definition 2.3.1. Observe that for any $x \in \mu^+$, the Bayes optimal classifier and the neighborhood preserving Bayes optimal both have the same output, and furthermore the neighborhood preserving Bayes gives this output (by definition) throughout the entirety of V_x, the neighborhood preserving robustness region of x. It follows that the neighborhood preserving Bayes optimal has optimal astuteness, as desired. □

Proof. (Theorem 2.3.4) Let $\mathscr{D} = (\mu, \eta)$ be a data distribution, and assume towards a contradiction that there exists classifier f which has maximal astuteness with respect towards some set of robustness regions $\mathscr{U} = \{U_x\}$ such that $V_x \subseteq U_x$ for all x. The key observation is that because f has maximal astuteness, we must have $f(x) = g(x)$ for almost all points $x \sim \mu$ (where g is the Bayes optimal classifier). Furthermore, for those values of x, we must have g be robust at x (meaning it uniformly outputs the same output through U_x).

In order for U_x to be strictly larger than V_x for some x, it *necessarily* must intersect with $U_{x'}$ for some x' with $g(x') \neq g(x)$, and this is what causes the contradiction: f cannot be astute at both x and x' if they are differently labeled and their robustness regions intersect. □

B.2.2 Proof of Theorem 2.4.1

Let $\mathscr{D} = (\mu, \eta)$ be the distribution with μ being the uniform distribution over $[0,1]$ and $\eta : [0,1] \to [0,1]$ be $\eta(x) = x$. For example, if $(x,y) \sim \mathscr{D}$, then $\Pr[y = 1 | x = 0.3] = 0.3$.

We desire to show that k_n-nearest neighbors is not neighborhood consistent with respect to $\mathscr{D}$. We begin with the following key lemma.

Lemma B.2.1. *For any $n > 0$, let f_n denote the k_n-nearest neighbor classifier learned from $S \sim \mathscr{D}^n$. There exists some constant $\Delta > 0$ such that for all sufficiently large n, with probability at least $\frac{1}{2}$ over $S \sim \mathscr{D}^n$, there exists $x \in [0,1]$ with $\frac{1}{2} - \Delta \leq x \leq \frac{1}{2} - \frac{3\Delta}{4}$ and $f_n(x) = +1$.*

Proof. Let C be a constant such that $k_n \leq C \log n$ for all $2 \leq n < \infty$. Set Δ as

$$\frac{1}{2} \log_2 \frac{1}{1-2\Delta} + \frac{1}{2} \log_2 \frac{1}{1+2\Delta} < \frac{1}{C}. \tag{B.1}$$

Let $A \subset [0,1]$ denote the interval $[\frac{1}{2} - \Delta, \frac{1}{2} - \frac{3\Delta}{4}]$. For $S \sim \mathscr{D}^n$, with high probability, there exist at least $\frac{\Delta n}{8}$ instances x_i that are in A. Let us relabel these x_i as $x_1, x_2, \ldots, x_m$ as

$$\frac{1}{2} - \Delta \leq x_1 < x_2 < \cdots < x_m \leq \frac{1}{2} - \frac{3\Delta}{4}.$$

Next, suppose that for some i, at least half of $y_i, y_{i+1}, \ldots, y_{i+k_n-1}$ are $+1$. Then it follows that $f_n(x) = +1$ for $x = \frac{x_{i+k_n}+x_i}{2}$ because the k_n nearest neighbors of x are precisely $x_i, x_{i+1}, \ldots x_{i+k_n-1}$ (as a technical note we make x just slightly smaller to break the tie between x_i and x_{i+k_n}). To lower bound the probability that this occurs for some i, we partition $y_1, y_2, \ldots y_m$ into at least $\frac{m}{2k_n}$ disjoint groups each containing k_n consecutive values of y_i. We then bound the probability that each group will have at least $k_n/2$ +1s.

Consider any group of k_n y_is. We have that $\Pr[y_i] = +1 = \eta(x_i) = x_i \geq \frac{1}{2} - \Delta$. Since the variables y_i are independent (even conditioning on x_i), it follows that the probability that at least half of them are $+1$ is at least $\Pr[\text{Bin}(k_n, \frac{1}{2} - \Delta) \geq \frac{k_n}{2}]$. For simplicity, assume that k_n is even. Then using a standard lower bound for the tail of a binomial distribution (see, for example, Lemma 4.7.2 of [80]), we have that

$$\Pr[\text{Bin}(k_n, \frac{1}{2} - \Delta) \geq \frac{k_n}{2}] \geq \frac{1}{\sqrt{2k_n}} \exp(-k_n D(\frac{1}{2}||(\frac{1}{2} - \Delta)),$$

where $D(\frac{1}{2}||(\frac{1}{2} - \Delta)) = \frac{1}{2}\log_2 \frac{1}{1-2\Delta} + \frac{1}{2}\log_2 \frac{1}{1+2\Delta}$.

To simplify notation, let $D_\Delta = D(\frac{1}{2}||(\frac{1}{2} - \Delta))$. Then because we have $\frac{m}{2k_n}$ independent groups of y_is, we have that

$$\begin{aligned}
\Pr_{S\sim\mathscr{D}^n}[\exists x \in [\frac{1}{2} - \Delta, \frac{1}{2} - \frac{3\Delta}{4}] \text{ s.t. } f_n(x) = +1] &\geq 1 - (1 - \frac{1}{\sqrt{2k_n}} \exp(-k_n D_\Delta))^{\frac{m}{2k_n}} \\
&\geq 1 - \exp(-\frac{m}{2k_n\sqrt{2k_n}} e^{-k_n D_\Delta}) \\
&\geq 1 - \exp(-\frac{n\Delta}{(16C\log n)^{3/2}} e^{-CD_\Delta \log n}),
\end{aligned}$$

with the inequalities holding because $m \geq \frac{n\Delta}{8}$ and $k_n \leq C\log n$. By equation B.1, $CD_\Delta < 1$. Therefore, $\lim_{n\to\infty} \frac{n}{(2C\log n)^{3/2}} e^{-CD_\Delta \log n} = \infty$, which implies that for n sufficiently large,

$$\Pr_{S\sim\mathscr{D}^n}[\exists x \in [\frac{1}{2} - \Delta, \frac{1}{2} - \frac{3\Delta}{4}] \text{ s.t. } f_n(x) = +1] \geq \frac{1}{2},$$

as desired. □

We now complete the proof of Theorem 2.4.1.

Proof. (Theorem 2.4.1) Let Δ be as described in Lemma B.2.1, and let $\kappa = \frac{1}{2}$. For all $x < \frac{1}{2}$, we have that $[x, \frac{2x}{3} + \frac{1}{6}] \subseteq V_x^\kappa$. This is because we can easily verify that all points inside that interval are closer to x than they are to $\frac{1}{2}$ (and consequently all points in $\mu^+ \cup \mu^{1/2}$) by factor of 2. It follows that for all $x \in [\frac{1}{2} - \frac{7\Delta}{8}, \frac{1}{2} - \Delta]$,

$$[\frac{1}{2} - \Delta, \frac{1}{2} - \frac{3\Delta}{4}] \subseteq V_x^\kappa.$$

However, applying Lemma B.2.1, we know that with probability at least $\frac{1}{2}$, there exists some point $x' \in [\frac{1}{2} - \Delta, \frac{1}{2} - \frac{3\Delta}{4}]$ such that $f_n(x') = +1$. It follows that with probability at least $\frac{1}{2}$, f_n lacks astuteness at *all* $x \in [\frac{1}{2} - \frac{7\Delta}{8}, \frac{1}{2} - \Delta]$. Since this set of points has total probability mass $\Delta/8$, it follows that with probability at least $\frac{1}{2}$, there is a fixed gap between $A_{\mathscr{V}^\kappa}(f_n, \mathscr{D})$ and $A(g, \mathscr{D})$ (as they differ in a region of probability mass at least $\Delta/8$). This implies that k_n-nearest neighbors is not neighborhood consistent. □

B.2.3 Proof of Theorem 2.4.4

Let $\mathscr{D} = (\mu, \eta)$ is a distribution over $\mathbb{R}^d \times \{\pm 1\}$. We will use the following notation: let $\mathscr{D}^+ = \{x : \eta(x) > \frac{1}{2}\}$, $\mathscr{D}^- = \{x : \eta(x) < \frac{1}{2}$ and $\mathscr{D}_{1/2} = \{x : \eta(x) = \frac{1}{2}\}$. In particular, we have that $\mathscr{D}^+ = \mu^+, \mathscr{D}^- = \mu^-$ and $\mathscr{D}_{1/2} = \mu^{1/2}$. This notation serve will be convenient throughout this section since it allows us to avoid overloading the symbol μ.

To show that an algorithm is neighborhood consistent with respect to $\mathscr{D}$, we must show that for any $0 < \kappa < 1$, the astuteness with respect to $\mathscr{V}^\kappa$ converges towards the accuracy of the Bayes optimal. To this end, we fix any $0 < \kappa < 1$ and consider $\mathscr{V}^\kappa$.

For our proofs, it will be useful to have the additional assumption that the robustness regions, V_x^κ are *closed*. To obtain this, we let $\mathscr{U} = \{U_x\}$ where $U_x = \overline{V_x^\kappa}$. Each U_x is the closure

of the corresponding V_x^κ, and in particular we have $V_x^\kappa \subset U_x$. Because of this, it will suffice for us to consider $A_{\mathscr{U}}$ as opposed to $A_{\mathscr{V}^\kappa}$ since $A_{\mathscr{U}}(f,\mathscr{D}) \leq A_{\mathscr{V}^\kappa}(f,\mathscr{D})$ for all classifiers f.

We now begin by first proving several useful properties of $\mathscr{U}$ that we will use throughout this entire section.

Lemma B.2.2. *The collection of sets $\mathscr{U} = \{U_x\}$ defined as $U_x = \overline{V_x^\kappa}$ satisfies the following properties.*

1. *U_x is closed for all x.*
2. *if $x \in \mathscr{D}^+$, for all $x' \in U_x$, $\rho(x,x') < \rho(\mathscr{D}^+ \cup \mathscr{D}_{1/2}, x')$.*
3. *if $x \in \mathscr{D}^-$, for all $x' \in U_x$, $\rho(x,x') < \rho(\mathscr{D}^- \cup \mathscr{D}_{1/2}, x')$.*
4. *$U_x = \{x\}$ for all $x \in \mathscr{D}_{1/2}$.*
5. *U_x is bounded for all x.*

Here $\mu^+, \mu^-, \mu^{1/2}$ are as described in Definition 2.3.1.

Proof. Property (1) is given the by definition, and properties (2), (3) follow from the fact that κ is strictly less than 1. In particular, the distance function ρ is continuous and consequently all limit points of a set have distances that are limits of distances within the set. Property (4) is since $V_x^\kappa = \{x\}$ for all $x \in \mathscr{D}_{1/2}$.

Finally, property (5) follows from the fact that $\kappa < 1$. As x gets arbitrarily far away from x the ratio of its distance to x with its distance to μ^- gets arbitrarily close to 1, and consequently there is some maximum radius R so that $V_x^\kappa \subset B(x,R)$. Since $B(x,R)$ is closed, it follows that $U_x \subset B(x,R)$ as well. □

Next, fix W as a weight function and t_n is a sequence of positive integers such that the conditions of Theorem 2.4.4 hold, that is:

1. W is consistent (with resp. to accuracy) with resp. to $\mathscr{D}$.

2. For any $0 < p < 1$, $\lim_{n\to\infty} E_{S\sim\mathscr{D}^n}[\sup_{x\in\mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p(x)}] = 0$.

3. $\lim_{n\to\infty} E_{S\sim D^n}[t_n \sup_{x\in\mathbb{R}^d} w_i^S(x)] = 0$.

4. $\lim_{n\to\infty} E_{S\sim D^n} \frac{\log T(W,S)}{t_n} = 0$.

Finally, we will also make the additional assumption that $\mathscr{D}$ has infinite support. Cases where $\mathscr{D}$ has finite support can be somewhat trivially handled: when the sample size goes to infinity, we will have perfect labels for every point in the support, and consequently condition 2. will ensure that any $x' \in V_x^\kappa$ is labeled according to the label of x.

We also use the following notation. For any classifier $f : \mathbb{R}^d \to \{\pm 1\}$, we let

$$\mathscr{D}_f^+ = \{x : f(x' = +1 \text{ for all } x' \in U_x\}, \text{ and } \mathscr{D}_f^- = \{x : f(x' = -1 \text{ for all } x' \in U_x\}. \tag{B.2}$$

These sets represent the examples that f robustly labels as $+1$ and -1 respectively. These sets are useful since they allows us to characterize the astuteness of f, which we do with the following lemma.

Lemma B.2.3. *For any classifier $f : \mathbb{R}^d \to \{\pm 1\}$, we have*

$$A_{\mathscr{U}}(f,\mathscr{D}) \geq A(g,\mathscr{D}) - \mu(\mathscr{D}^+ \setminus \mathscr{D}_f^+) - \mu(D^- \setminus \mathscr{D}_f^-),$$

where g denotes the Bayes optimal classifier.

Proof. By property 4 of Lemma B.2.2, $U_x = \{x\}$ for all $x \in \mathscr{D}_{1/2}$. Consequently, if $x \in \mathscr{D}_{1/2}$, there is a $\frac{1}{2}$ chance that any classifier is astute at (x,y). Using this along with the definition of astuteness, we see that

$$\begin{aligned} A_{\mathscr{U}}(f,\mathscr{D}) &= \Pr_{(x,y)\sim\mathscr{D}}[f(x') = y \text{ for all } x' \in U_x] \\ &= \Pr_{(x,y)\sim\mathscr{D}}[y = +1 \text{ and } x \in (D^+ \cap D_f^+)] + \Pr_{(x,y)\sim\mathscr{D}}[y = -1 \text{ and } x \in (D^- \cap D_f^-)] + \frac{1}{2}\Pr_{(x,y)\sim\mathscr{D}}[x \in \mathscr{D}_{1/2}] \end{aligned}$$

However, observe by the definitions of $\mathscr{D}^+, \mathscr{D}^-$ and $\mathscr{D}_{1/2}$ that

$$A(g,\mathscr{D}) = \Pr_{(x,y)\sim\mathscr{D}}[y=+1 \text{ and } x\in D^+] + \Pr_{(x,y)\sim\mathscr{D}}[y=-1 \text{ and } x\in D^-] + \frac{1}{2}\Pr_{(x,y)\sim\mathscr{D}}[x\in\mathscr{D}_{1/2}].$$

Substituting this, we find that

$$\begin{aligned} A_{\mathscr{U}}(f,\mathscr{D}) &\geq A(g,\mathscr{D}) - \Pr_{(x,y)\sim\mathscr{D}}[x\in(D^+\setminus D_f^+)] - \Pr_{(x,y)\sim\mathscr{D}}[x\in(D^-\setminus D_f^-)] \\ &= A(g,\mathscr{D}) - \mu(\mathscr{D}^+\setminus\mathscr{D}_f^+) - \mu(D^-\setminus\mathscr{D}_f^-), \end{aligned}$$

as desired. □

Lemma B.2.3 shows that to understand how W_S converges in astuteness, it suffices to understand how the regions $\mathscr{D}^+_{W_S}$ and $\mathscr{D}^-_{W_S}$ converge towards D^+ and D^- respectively. This will be our main approach for proving Theorem 2.4.4. Due to the inherent symmetry between $+$ and $-$, we will focus on showing how the region $\mathscr{D}^+_{W_S}$ converges towards D^+. The case for $-$ will be analogous. To that end, we have the following key definition.

Definition B.2.4. *Let $p,\Delta>0$. We say $x\in\mathscr{D}^+$ is (p,Δ)-**covered** if for all $x'\in U_x$ and for all $x''\in B(x',r_p(x'))\cap supp(\mu)$, $\eta(x'')>\frac{1}{2}+\Delta$. Here r_p denotes the probability radius (Definition 2.4.2). We also let $\mathscr{D}^+_{p,\Delta}$ denote the set of all $x\in\mathscr{D}^+$ that are (p,Δ)-covered.*

If x is (p,Δ)-covered, it means that for all $x'\in U_x$, there is a set of points with measure p around x' that are both close to x', and likely (with at least probability $\frac{1}{2}+\Delta$) to be labeled as $+1$. Our main idea will be to show that if x is (p,Δ) covered and n is sufficiently large, x is likely to be in $\mathscr{D}^+_{W_S}$.

We begin this process by first showing that all x are (p,Δ)-covered for some p,Δ. To do so, it will be useful to have one more piece of notation which we will also use throughout the rest of the section. We let

$$\mathscr{D}^-_{1/2} = \mathscr{D}^-\cup\mathscr{D}_{1/2} = supp(\mu)\setminus\mathscr{D}^+.$$

This set will be useful, since Lemma B.2.2 implies that for all $x \in \mathscr{D}^+$ and for all $x' \in U_x$, $\rho(x,x') < \rho(\mathscr{D}^-_{1/2}, x')$. We now return to showing that all x are $(p,\Delta$-covered for some p,Δ.

Lemma B.2.5. *For any $x \in \mathscr{D}^+$, there exists $p,\Delta > 0$ such that x is (p,Δ)-covered.*

Proof. Fix any x. Let $f : U_x \to \mathbb{R}$ be the function defined as $f(x') = \rho(x', \mathscr{D}^-_{1/2}) - \rho(x',x)$. Observe that f is continuous. By assumption, U_x is closed and bounded, and consequently must attain its minimum. However, by Lemma B.2.2, we have that $f(x') > 0$ for all $x' \in U_x$. it follows that $\min_{x' \in U_x} f(x') = \gamma$ where $\gamma > 0$.

Next, let $p = \mu(B(x,\gamma/2))$. $p > 0$ since $x \in supp(\mu)$. Observe that for any $x' \in U_x$, $r_p(x') \leq \rho(x,x') + \gamma/2$, where, $r_p(x')$ denotes the probability radius of x'. This is because $B(x', (\rho(x,x') + \gamma/2))$ contains $B(x,\gamma/2)$ which has probability mass p. It follows that for any $x' \in U_x$, $\rho(x', \mathscr{D}^-_{1/2}) \geq r_p(x') + \gamma/2$. Motivated by this observation, let A be the region defined as

$$A = \bigcup_{x' \in U_x} B(x', r_p(x')).$$

Then by our earlier observation, we have that $\rho(A, \mathscr{D}^-_{1/2}) \geq \frac{\gamma}{2}$. Since distance is continuous, it follows that $\rho(\overline{A}, \mathscr{D}^-_{1/2}) \geq \frac{\gamma}{2}$ as well, where $\overline{A}$ denotes the closure of A.

This means that for any $x'' \in \overline{A} \cap supp(\mu)$, $\eta(x'') > \frac{1}{2}$, since otherwise $\rho(\overline{A}, \mathscr{D}^-_{1/2})$ would equal 0 (as the two sets would literally intersect). Finally, $supp(\mu)$ is a closed set (see Appendix B.3.1), and thus $\overline{A} \cap supp(\mu)$ is closed as well. Since η is continuous (by assumption from Definition 2.3.6), it follows that η must maintain its minimum value over $\overline{A} \cap supp(\mu)$. It follows that there exists $2\Delta > 0$ such that $\eta(x'') \geq \frac{1}{2} + 2\Delta > \frac{1}{2} + \Delta$ for all $x'' \in \overline{A} \cap supp(\mu)$.

Finally, by the definition of A, for all $x' \in U_x$, $B(x', r_p(x')) \subset A$. It consequently follows from the definition that x is (p,Δ)-covered, as desired. □

While the previous lemma show that some p,Δ cover any $x \in \mathscr{D}^+$, this does not necessarily mean that there are some fixed p,Δ that cover *all* $x \in \mathscr{D}^+$. Nevertheless, we can show that

this is almost true, meaning that there are some p, Δ that cover *most* $x \in \mathscr{D}^+$. Formally, we have the following lemma.

Lemma B.2.6. *For any $\varepsilon > 0$, there exists p, Δ such that $\mu(\mathscr{D}^+ \setminus \mathscr{D}^+_{p,\Delta}) < \varepsilon$, where $\mathscr{D}^+_{p,\Delta}$ is as defined in Definition B.2.4.*

Proof. Observe that if x is (p,Δ)-covered, then it is also (p',Δ')-covered for any $p' < p$ and $\Delta' < \Delta$. This is because $B(x', r_{p'}(x')) \subset B(x', r_p(x))$ and $\frac{1}{2} + \Delta > \frac{1}{2} + \Delta'$. Keeping this in mind, define

$$\mathscr{A} = \{\mathscr{D}^+_{1/i,1/j} : i, j \in \mathbb{N}\}.$$

For any $x \in \mathscr{D}^+$, by Lemma B.2.5 and our earlier observation, there exists $A \in \mathscr{A}$ such that $x \in A$. It follows that $\cup_{A \in \mathscr{A}} A = \mathscr{D}^+$. By applying Lemma B.3.1, we see that there exists a finite subset of $\mathscr{A}$, $\{A_1, \ldots, A_m\}$ such that

$$\mu(A_1 \cup \cdots \cup A_m\}) > \mu(\mathscr{D}^+) - \varepsilon.$$

Let $A_k = \mathscr{D}^+_{1/i_k,1/j_k}$ for $1 \leq k \leq m$. From our previous observation once again, we see that $\cup A_i \subset \mathscr{D}^+_{1/I,1/J}$ where $I = \max(i_k)$ and $J = \max(j_k)$. It follows that setting $p = 1/I$ and $\Delta = 1/J$ suffices. □

Recall that our overall goal is to show that if x is (p,Δ)-covered, n is sufficiently large, then x is very likely to be in $\mathscr{D}^+_{W_S}$ (defined in equation B.2). To do this, we will need to find sufficient conditions on S for x to be in W_S. This requires the following definitions, that are related to *splitting numbers* (Definition 2.4.3).

Definition B.2.7. *Let $x \in \mathbb{R}^d$ be a point, and let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set sampled from $\mathscr{D}^n$. For $0 \leq \alpha$, $0 \leq \beta \leq 1$, and $0 < \Delta < \frac{1}{2}$, we define*

$$W^{\Delta,S}_{x,\alpha,\beta} = \{i : \rho(x, x_i) \leq \alpha, w^S_i(x) \geq \beta, \eta(x_i) > \frac{1}{2} + \Delta\}.$$

Definition B.2.8. *Let $0 < \Delta < \frac{1}{2}$, and let $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ be a training set sampled from $\mathscr{D}^n$. Then we let*

$$W^{\Delta,S} = \{W^{\Delta,S}_{x,\alpha,\beta} : x \in \mathbb{R}^d, 0 \leq \alpha, 0 \leq \beta \leq 1\}.$$

These convoluted looking sets will be useful for determining the behavior of W_s at some $x \in \mathscr{D}^+_{p,\Delta}$. Broadly speaking, the idea is that if every set of indices $R \subset W^{\Delta,S}$ is relatively well behaved (i.e. the number of y_is that are $+1$ is close to $(|R|(\frac{1}{2}+\Delta)$, the expected amount), then $W_s(x') = +1$ for all $x' \in U_x$. Before showing this, we will need a few more lemmas.

Lemma B.2.9. *Fix any $\delta > 0$ and let $0 < \Delta < \frac{1}{2}$. There exists N such that for all $n > N$ the following holds. With probability $1-\delta$ over $S \sim \mathscr{D}^n$, for all $R \in W^{\Delta,S}$ with $|R| > t_n$, $\frac{1}{|R|}\sum_{i\in R} y_i \geq \Delta$*

Proof. The key idea is to observe that the set $W^{\Delta,S}$ and the value $T(W,S)$ are completely determined by $\{x_1, \ldots, x_n\}$. This is because weight functions choose their weights only through dependence on $x_1, \ldots, x_n$. Consequently, we can take the equivalent formulation of first drawing $x_1, \ldots, x_n \sim \mu^n$, and then drawing y_i independently according to $y_i = 1$ with probability $\eta(x_1)$ and 0 with probability $1 - \eta(x_i)$. In particular, we can treat $y_1, \ldots, y_n$ as independent from $W^{\Delta,S}$ and $T(W,S)$ conditioning on $x_1, \ldots, x_n$.

Fix any $x_1, \ldots, x_n$. First, we see that $|W^{\Delta,S}| \leq T(W,S)$. This is because $W^{\Delta,S}_{x,\alpha,\beta}$ is a subset that is uniquely defined by $W_{x,\alpha,\beta}$ (see Definitions B.2.7 and 2.4.3). Second, for any $R \in W^{\Delta,S}$, observe that for all $i \in R$, y_i is a binary variable in $[-1,1]$ with expected value at least $(\frac{1}{2}+\Delta) - (\frac{1}{2}-\Delta) = 2\Delta$ (again by the definition). It follows that if $|R| \geq t_n$, by Hoeffding's inequality

$$\Pr_{y_1 \ldots y_n}[\sum_{i\in R} y_i < \Delta] \leq \exp\left(-\frac{2|R|^2\Delta^2}{4|R|}\right) \leq \exp\left(-\frac{t_n\Delta^2}{2}\right).$$

Since there at most $T(W,S)$ sets R, it follows that

$$\Pr_{y_1 \ldots y_n}[\sum_{i\in R} y_i < \Delta \text{ for some } R \in W^{\Delta,S} \text{ with } |R| > t_n] \leq T(W,S)\exp\left(-\frac{t_n\Delta^2}{2}\right).$$

However, by condition 4. of Theorem 2.4.4, it is not difficult to see that this quantity has expectation that tends to 0 as $n \to \infty$ (unless $T(W,S)$ uniformly equals 1, but this degenerate case can easily be handled on its own). Thus, for any $\delta > 0$, it follows that there exists N such that for all $n > N$, with probability at least $1 - \frac{\delta}{2}$, $T(W,S)\exp\left(-\frac{t_n\Delta^2}{2}\right) \leq \frac{\delta}{2}$. This value of N consequently suffices for our lemma. □

We now relate $\mathscr{D}^+_{W_S}$ (Equation B.2) to $W^{\Delta,S}$ as well as the conditions of Theorem 2.4.4.

Lemma B.2.10. *Let $S = \{(x_1,y_1),\ldots,(x_n,y_n)\}$ and let $0 < \Delta \leq \frac{1}{2}$ and $0 < p < 1$ such that the following conditions hold.*

1. *For all $R \in W^{\Delta,S}$ with $|R| > t_n$, $\frac{1}{|R|}\sum_{i\in R} y_i \geq \Delta$.*
2. $\sup_{x\in\mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)} < \frac{\Delta}{5}$.
3. $t_n \sup_{x\in\mathbb{R}^d} w_i^S(x) < \frac{\Delta}{5}$.

Then $\mathscr{D}^+_{p,\Delta} \subseteq \mathscr{D}^+_{W_S}$.

Proof. Let $x \in \mathscr{D}^+_{p,\Delta}$, and let $x' \in U_x$ be arbitrary. It suffices to show that $W_S(x') = +1$ (as x, x' were arbitrarily chosen). From the definition of W_S, this is equivalent to showing that $\sum_1^n w_i^S(x')y_i > 0$. Thus, our strategy will be to lower bound this sum using the conditions given in the lemma statement.

We first begin by simplifying notation. Since S and x' are both fixed, we use w_i to denote $w_i^S(x')$. Since n is fixed, we will also use t to denote t_n. Next, suppose that $|\{x_1,\ldots,x_n\} \cap B(x', r_p(x'))| = k$. Without loss of generality, we can rename indices such that $\{x_1,\ldots,x_n\} \cap B(x', r_p(x')) \cap B(x', r_p(x')) = \{x_1,\ldots,x_k\}$, and $w_1 \geq w_2 \geq \cdots \geq w_k$.

Let $Y_j = \sum_{i=1}^j y_i$. Our main idea will be to express the sum in terms of these Y_js as

follows.

$$\begin{aligned}\sum_1^n w_i y_i &= \sum_1^k w_i y_i + \sum_{k+1}^n w_i y_i \\ &= w_k Y_k + (w_{k-1} - w_k) Y_{k-1} + \cdots + (w_{t+1} - w_{t+2}) Y_{t+1} + \sum_{i=1}^t (w_i - w_{t+1}) y_i + \sum_{k+1}^n w_i y_i \\ &= \underbrace{w_k Y_k + \sum_{i=t+1}^{k-1} (w_i - w_{i+1}) Y_i}_{\alpha} + \underbrace{\sum_{i=1}^t (w_i - w_{t+1}) y_i}_{\beta} + \underbrace{\sum_{k+1}^n w_i y_i}_{\tau}.\end{aligned}$$

We now bound α, β and τ in terms of Δ by using the conditions given in the lemma. We begin with β and τ, which are considerably easier to handle.

For β, we have that

$$\beta = \sum_{i=1}^t (w_i - w_{t+1}) y_i \geq \sum_{i=1}^t (w_i - w_{t+1})(-1) \geq -t w_1.$$

By condition 2 of the lemma, we see that $t w_1 < \frac{\Delta}{5}$, which implies that $\beta \geq -\frac{\Delta}{5}$.

For γ, we have that $\gamma = \sum_{k+1}^n w_i y_i \geq -\sum_{k+1}^n w_i$. However, for all $k+1 \leq i \leq n$, by definition of k, $\rho(x', x_i) > r_p(x')$. It follows from condition 3 of the lemma that $\gamma \geq -\frac{\Delta}{5}$.

Finally, we handle α. Recall that x is (p, Δ)-covered. It follows that for all $x'' \in supp(\mu) \cap B(x', r_p(x'))$, $\eta(x'') > \frac{1}{2} + \Delta$. Thus, by the definition of k, $\eta(x_i) > \frac{1}{2} + \Delta$ for $1 \leq i \leq k$. It follows that if $w_i > w_{i+1}$ or $i = k$, then

$$\begin{aligned}W^{\Delta,S}_{x', r_p(x'), w_i} &= \{j : \rho(x', x_j) \leq r_p(x'), w_j \geq w_i, \eta(x_j) > \frac{1}{2} + \Delta\} \\ &= \{1, \ldots, i\}.\end{aligned}$$

This implies that $\{1, \ldots, i\} \in W^{\Delta,S}$, and consequently that $Y_i \geq i\Delta$, from condition 1 of the lemma. It follows that for all $t < i \leq k$, $(w_i - w_{i+1}) Y_i \geq i(w_i - w_{i+1})\Delta$, and that $w_k Y_k \geq k w_k \Delta$.

Substituting these, we find that

$$\begin{aligned}
\alpha &= w_k Y_k + \sum_{i=t+1}^{k-1} (w_i - w_{i+1}) Y_i \\
&\geq k w_k \Delta + \sum_{i=t+1}^{k-1} i(w_i - w_{i+1})\Delta \\
&= w_k\Delta + w_{k-1}\Delta + \cdots + w_{t+1}\Delta + (t+1)w_{t+1}\Delta. \\
&\geq (1 - \sum_{1}^{t} w_i - \sum_{k+1}^{n} w_i)\Delta \\
&\geq (1 - \frac{2\Delta}{5})\Delta \\
&\geq (\frac{4\Delta}{5}),
\end{aligned}$$

with the last inequalities holding from the arguments given for β and γ along with the fact that $0 < \Delta \leq \frac{1}{2}$. Finally, substituting these, we find that $\alpha + \beta + \gamma \geq \frac{4\Delta}{5} - \frac{2\Delta}{5} = \frac{2\Delta}{5} > 0$, as desired. □

We are now ready to prove the key lemma that forms one half of the main theorem (the other half corresponding to $\mathscr{D}^{-}_{W_S}$).

Lemma B.2.11. *Let $\delta, \varepsilon > 0$. There exists N such that for all $n > N$, with probability $1 - \delta$ over $S \sim \mathscr{D}^n$, $\mu(\mathscr{D}^{+} \setminus \mathscr{D}^{+}_{W_S}) < \varepsilon$.*

Proof. First, by Lemma B.2.6, let $0 < p$ and $0 < \Delta$ be such that $\mu(\mathscr{D}^{+} \setminus \mathscr{D}^{+}_{p,\Delta}) < \varepsilon$. By combining Lemma B.2.9, condition 3 of Theorem 2.4.4, and condition 2 of Theorem 2.4.4 respectively, we see that there exists N such that for all $n > N$, the following hold:

1. With probability at least $1 - \frac{\delta}{3}$ over $S \sim \mathscr{D}^n$, for all $R \in W^{\Delta,S}$ with $|R| > t_n$, $\frac{1}{|R|}\sum_{i \in R} y_i \geq \Delta$.
2. With probability at least $1 - \frac{\delta}{3}$ over $S \sim \mathscr{D}^n$, $\sup_{x \in \mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)} < \frac{\Delta}{5}$.
3. With probability at least $1 - \frac{\delta}{3}$ over $S \sim \mathscr{D}^n$, $t_n \sup_{x \in \mathbb{R}^d} w_i^S(x) < \frac{\Delta}{5}$.

By a union bound, this implies that p, Δ, S satisfy the conditions of Lemma B.2.10 with probability at least $1 - \delta$. Thus, applying the Lemma, we see that with probability $1 - \delta$, $\mathscr{D}^{+}_{p,\Delta} \subset \mathscr{D}^{+}_{W_S}$. This

immediately implies our claim. □

By replicating all of the work in this section for $\mathscr{D}^-$ and $\mathscr{D}^-_{p,\Delta}$, we can similarly show the following:

Lemma B.2.12. *Let $\delta, \varepsilon > 0$. There exists N such that for all $n > N$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$, $\mu(\mathscr{D}^- \setminus \mathscr{D}^-_{W_S}) < \varepsilon$.*

Combining these two lemmas with Lemma B.2.3 immediately implies that for all $\delta, \varepsilon > 0$, there exists N such that for all $n > N$, with probability $1-\delta$ over $S \sim \mathscr{D}^n$,

$$A_{\mathscr{U}}(W_S, \mathscr{D}) \geq A(g, \mathscr{D}) - \varepsilon.$$

Since $V_x^\kappa \subset U_x$ and since κ was arbitrary, this implies Theorem 2.4.4, which completes our proof.

B.2.4 Proof of Corollary 2.4.5

Recall that k_n-nearest neighbors can be interpreted as a weight function, in which $w_i^S(x) = \frac{1}{k_n}$ if x_i is one of the k_n closest points to x, and 0 otherwise. Therefore, it suffices to show that the conditions of Theorem 2.4.4 are met.

We let W denote the weight function associated with k_n-nearest neighbors.

Lemma B.2.13. *W is consistent.*

Proof. It is well known (for example [37]) that k_n-nearest neighbors is consistent for $\lim_{n\to\infty} k_n = \infty$ and $\lim_{n\to\infty} \frac{k_n}{n} = 0$. These can easily be verified for our case. □

Lemma B.2.14. *For any $0 < p < 1$, $\lim_{n\to\infty} \mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x\in\mathbb{R}^d} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)}] = 0$.*

Proof. It suffices to show that for n sufficiently large, all k_n-nearest neighbors of x are located inside $B(x, r_p(x))$ for all $x \in \mathbb{R}^d$. We do this by using a VC-dimension type argument to show that all balls $B(x,r)$ contain a number of points from $S \sim \mathscr{D}^n$ that is close to their expectation.

For $x \in \mathbb{R}^d$ and $r \geq 0$, let $f_{x,r}$ denote the $0-1$ function defined as $f_{x,r}(x') = 1_{x' \in B(x,r)}$. Let $F = \{f_{x,r} : x \in \mathbb{R}^d, r \geq 0\}$ denote the class of all such functions. It is well known that the VC dimension of F is at most $d+2$.

For $f \in F$, let $\mathbb{E}f$ denote $\mathbb{E}_{(x',y)\sim\mathscr{D}} f(x')$ and $\mathbb{E}_n f$ denote $\frac{1}{n}\sum_1^n f(x_i)$, where $\mathbb{E}_n f$ is defined with respect to some sample $S \sim \mathscr{D}^n$. By the standard generalization result of Vapnik and Chervonenkis (see [81] for a proof), we have that with probability $1-\delta$ over $S \sim \mathscr{D}^n$,

$$-\beta_n \sqrt{\mathbb{E}f} \leq \mathbb{E}f - \mathbb{E}_n f \leq \beta_n \sqrt{\mathbb{E}f} \tag{B.3}$$

holds for all $f \in F$, where $\beta_n = \sqrt{(4/n)((d+2)\ln 2n + \ln(8/\delta)}$.

Suppose n is sufficiently large so that $\beta_n \leq \frac{p}{2}$ and $\frac{k_n}{n} < \frac{p}{2}$, and suppose that equation B.3 holds. Pick any $x \in \mathbb{R}^d$ and consider $f_{x,r}$ where $r > r_p(x)$. This implies $\mathbb{E}f_{x,r} \geq p$. Then by equation B.3, we see that $\mathbb{E}_n f \geq \frac{p}{2}$. This implies that all k_n nearest neighbors of x are in the ball $B(x,r)$, and that consequently $\sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r} = 0$. Because this holds for all x, r with $x \in \mathbb{R}^d$ and $r > r_p(x)$, it follows that equation 2 implies that

$$\sup_{x \in X} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)} = 0.$$

Because equation B.3 holds with probability at least $1-\delta$, and δ can be made arbitrarily small, the desired claim follows. □

Let $t_n = \sqrt{dk_n \log n}$.

Lemma B.2.15. $\lim_{n\to\infty} E_{S\sim D^n}[t_n \sup_{x\in\mathbb{R}^d} w_i^S(x)] = 0$.

Proof. Let $S \sim \mathscr{D}^n$. By the definition of k_n nearest neighbors, $\sup_{x\in\mathbb{R}^d} w_i^S(x) = \frac{1}{k_n}$. Therefore, $t_n \sup_{x\in\mathbb{R}^d} w_i^S(x) = \sqrt{\frac{d\log n}{k_n}}$. By assumption 2. of corollary 2.4.5, $\lim_{n\to\infty} \frac{d\log n}{k_n} = 0$, which implies that

$$\lim_{n\to\infty} \mathbb{E}_{S\sim D^n}[t_n \sup_{x\in\mathbb{R}^d} w_i^S(x)] = \lim_{n\to\infty} \sqrt{\frac{d\log n}{k_n}} = \lim_{n\to\infty} \frac{d\log n}{k_n} = 0,$$

as desired. □

Lemma B.2.16. $\lim_{n\to\infty} E_{S\sim D^n} \frac{\log T(W,S)}{t_n} = 0.$

Proof. For $S \sim \mathscr{D}^n$, recall that $T(W,S)$ was defined as

$$T(W,S)|\{W_{x,\alpha,\beta} : x \in \mathbb{R}^d, 0 \leq \alpha, 0 \leq \beta \leq 1\}|,$$

where $W_{x,\alpha,\beta}$ denotes

$$W_{x,\alpha,\beta} = \{i : \rho(x,x_i) \leq \alpha, w_i^S(x) \geq \beta\}.$$

Our goal will to be upper bound $\log T(W,S)$.

To do so, we first need a tie-breaking mechanism for k_n-nearest neighbors. For each $x_i \in S$, we independently sample $z_i \in [0,1]$ from the uniform distribution. We then tie break based upon the value of z_i, i.e. if $\rho(x,x_i) = \rho(x,x_j)$, we say that x_i is closer to x than x_j if $z_i < z_j$. With probability 1, no two values z_i, z_j will be equal, so this ensures that this method always works.

Let $A_{x,\alpha} = \{i : \rho(x,x_i) \leq \alpha\}$ and let $B_{x,c} = \{i : z_i \leq c\}$. The key observation is that for any α, β, $W_{x,\alpha,\beta} = A_{x,\alpha} \cap B_{x,c}$ for some value of c. This can be seen by noting that the nearest neighbors of x are uniquely determined by $\rho(x,x_i)$ and z_i. Therefore, it suffices to bound $|A = A_{x,\alpha} : x \in \mathbb{R}^d, \alpha \geq 0\}|$ and $|B = \{B_{x,c} : x \in \mathbb{R}^d, c \geq 0\}|$.

To bound $|A|$, observe that the set of closed balls in $\mathbb{R}^d$ has VC-dimension at most $d+2$. Thus by Sauer's lemma, there are at most $O(n^{d+2}$ subsets of $\{x_1, x_2, \ldots, x_n\}$ that can be obtained from closed balls. Thus $|A| \leq O(n^{d+2}$.

To bound $|B|$, we simply note that $B_{x,c}$ consists of all i for which $z_i \leq c$. Since the z_i can be sorted, there are at most $n+1$ such sets. Thus $|B| \leq n+1$.

Combining this, we see that $T(W,S) \leq |A||B| \leq O(n^{d+3})$. Finally, we see that

$$\lim_{n\to\infty} \frac{\log T(W,S)}{t_n} = \lim_{n\to\infty} \frac{O(d\log n)}{\sqrt{k_n d \log n}} = \lim_{n\to\infty} \sqrt{\frac{O(d\log n)}{k_n}} = 0,$$

with the last inequality holding by condition 2. of Corollary 2.4.5.

□

Finally, we note that Corollary 2.4.5 is an immediate consequence of the previous 4 lemmas as we can simply apply Theorem 2.4.4.

B.2.5 Proof of Corollary 2.4.6

Let W be a kernel classifier constructed from K and h_n such that the conditions of Corollary 2.4.6 hold: that is,

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

It suffices to show that the conditions of Theorem 2.4.4 are met for W. Before doing this, we will describe one additional assumption we make for this case.

Additional Assumption:

We assume that $\mathscr{D}, \mathscr{U}$ are such that there exists some compact set $\mathscr{X} \subset \mathbb{R}^d$ such that for all $x \in supp(\mu)$, $U_x \subset \mathscr{X}$. This is primarily for convenience: observe that any distribution can be approximated arbitrarily closely by distributions satisfying these properties (as each U_x is bounded by assumption). Importantly, because of this, we will note that it is possible for conditions 2. and 3. of Theorem 2.4.4 to be relaxed to taking supremums over $\mathscr{X}$ rather than $\mathbb{R}^d$. This is because in our proof, we only ever used these conditions in their restriction to $\bigcup_{x\in supp(\mu)} \bigcup x' \in U_x B(x', r_p(x'))$.

Using this assumption, we return to proving the corollary.

Lemma B.2.17. *W is consistent with respect to $\mathscr{D}$.*

Proof. Condition 1. of Corollary 2.4.6 imply that K is a regular kernel. This together with Condition 2. implies that W is consistent: a proof can be found in [12]. □

To verify the second condition, it will be useful to have the following definition.

Definition B.2.18. *For any $p, \varepsilon > 0$ and $x \in \mathscr{X}$, define r_p^ε as*

$$r_p^\varepsilon(x) = \sup\{r : \mu(B(x,r)) - \mu(B(x, r_p(x)) \leq \varepsilon\}.$$

Lemma B.2.19. *For any $p, \varepsilon > 0$, there exists a constant $c_p^\varepsilon > 1$ such that $\frac{r_p^\varepsilon(x)}{r_p(x)} \geq c_p^\varepsilon$ for all $x \in \mathscr{X}$, where we set $\frac{r_p^\varepsilon(x)}{r_p(x)} = \infty$ if $r_p(x) = 0$.*

Proof. The basic idea is to use the fact that $\mathscr{X}$ is compact. Our strategy will be to analyze the behavior of $\frac{r_p^\varepsilon(x)}{r_p(x)}$ over small balls $B(x_0, r)$ centered around some fixed x_0, and then use compactness to pick some finite set of balls $B(x_0, r)$. This must be done carefully because the function $x \to \frac{r_p^\varepsilon(x)}{r_p(x)}$ is not necessarily continuous.

Fix any $x_0 \in \mathscr{X}$. First, observe that $r_p^\varepsilon(x_0) > r_p(x_0)$. This is because $B(x_0, r_p(x_0)) = \cap_{r > r_p(x_0)} B(x_0, r)$, and consequently $\lim_{r \downarrow r_p(x_0)} \mu(B(x_0, r)) = \mu(B(x_0, r_p(x_)))$.

Next, define

$$s_p^\varepsilon(x) = \inf\{r : \mu(B(x, r_p(x)) - \mu(B(x,r)) \leq \varepsilon\}.$$

We can similarly show that $r_p(x_0) > s_p^\varepsilon(x_0)$.

Finally, define

$$r_0 = \frac{1}{3} \min(r_p^\varepsilon(x_0) - r_p(x_0), r_p(x_0) - s_p^\varepsilon(x_0)).$$

Consider any $x \in B^o(x_0, r_0)$ where B^o denotes the open ball, and let $\alpha = \rho(x_0, x)$. Then we have the following.

1. $r_p(x) \leq r_p(x_0) + \alpha$. This holds because $B(x, r_p(x_0) + \alpha)$ contains $B(x_0, r_p(x_0))$, which has probability mass at least p.

2. $r_p(x) \geq r_p(x_0) - \alpha$. This holds because if $r_p(x) < r_p(x_0) - \alpha$, then there would exists $r < r_p(x_0)$ such that $\mu(B(x_0, r)) \geq p$ which is a contradiction.

3. $B(x_0, s_p^\varepsilon(x_0)) \subset B(x, r_p(x))$. This is just a consequence of the definition of r_0 and the previous observation.

By the definitions of r_p^ε and s_p^ε, we see that $\mu(B(x_0, r_p^\varepsilon(x_0)) - \mu(B(x_0, s_p^\varepsilon(x_0)) \leq 2\varepsilon$. By the triangle inequality, $B(x, r_p^\varepsilon(x_0) - \alpha) \subset B(x_0, r_p^\varepsilon(x_0))$ and $B(x_0, s_p^\varepsilon(x_0)) \subset B(x, r_p(x))$. it follows that

$$\mu(B(x, r_p^\varepsilon(x_0) - \alpha)) - \mu(B(x, r_p(x))) \leq 2\varepsilon,$$

which implies that $r_p^{2\varepsilon}(x) \geq r_p^\varepsilon(x_0) - \alpha$. Therefore we have the for all $x \in B(x_0, r_0)$,

$$\frac{r_p^{2\varepsilon}(x)}{r_p(x)} \geq \frac{r_p^\varepsilon(x_0) - \alpha}{r_p(x_0) + \alpha} \geq \frac{2r_p^\varepsilon(x_0) + r_p(x_0)}{r_p^\varepsilon(x_0) + 2r_p(x_0)}.$$

Notice that the last expression is a constant that depends only on x_0, and moreover, since $r_p^\varepsilon(x_0) > r_p(x_0)$, this constant is strictly larger than 1. Let us denote this as $c(x_0)$. Then we see that $\frac{r_p^{2\varepsilon}(x)}{r_p(x)} \geq c(x_0)$ for all $x \in B^o(x_0, r_0)$.

Finally, observe that $\{B^o(x_0, r_0) : x_0 \in \mathscr{X}\}$ forms an open cover of $\mathscr{X}$ and therefore has a finite sub-cover C. Therefore, taking $c = \min_{B^o(x_0, r_0) \in C} c(x_0)$, we see that $\frac{r_p^{2\varepsilon}(x)}{r_p(x)} \geq c > 1$ for all $x \in \mathscr{X}$. Because ε was arbitrary, the claim holds. □

Lemma B.2.20. *For any* $0 < p < 1$, $\lim_{n\to\infty} \mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x\in\mathscr{X}} \sum_1^n w_i^S(x) 1_{\rho(x,x_i) > r_p(x)}] = 0$.

Proof. Fix $p > 0$, and fix any $\varepsilon, \delta > 0$. Pick n sufficiently large so that the following hold.

1. Let c_p^ε be as defined from Lemma B.2.19.

$$\sup_{x\in\mathscr{X}} \frac{K(c_p^\varepsilon r_p(x)/h_n)}{K(r_p(x)/h_n)} < \delta. \tag{B.4}$$

This is possible because of conditions 2. and 3. of Corollary 2.4.6, and because the function $x \to r_p(x)$ is continuous.

2. With probability at least $1-\delta$ over $S \sim \mathscr{D}^n$, for all $r > 0$, and $x \in \mathscr{X}$,

$$|\mu(B(x,r)) - \frac{1}{n}\sum_1^n 1_{x_i \in B(x,r)}| \leq \varepsilon. \tag{B.5}$$

This is possible because the set of balls $B(x,r)$ has VC dimension at most $d+2$.

We now bound $\mathbb{E}_{S\sim\mathscr{D}^n}[\sup_{x\in\mathscr{X}} \sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p(x)}]$ by dividing into cases where S satisfies and doesn't satisfy equation B.5.

Suppose S satisfies equation B.5. By condition 1. of Corollary 2.4.6, K is decreasing, and by Lemma B.2.19, $r_p^\varepsilon(x) \geq c_p^\varepsilon r_p(x)$. Therefore, we have that for any $x \in \mathscr{X}$,

$$\begin{aligned}\sum_1^n K(\rho(x,x_i)/h_n) 1_{\rho(x,x_i)\geq r_p^\varepsilon(x)} &\leq \sum_1^n K(c_p^\varepsilon r_p(x)/h_n) \\ &\leq n\delta K(r_p(x)/h_n)),\end{aligned}$$

where the second inequality comes from equation B.4.

Next, by the definition of $r_p^\varepsilon(x)$, we have that $\mu(B(x,r_p^\varepsilon(x)) - \mu(B(x,r_p(x))) \leq \varepsilon$. Therefore, by applying equation B.5 two times, we see that for any $x \in \mathscr{X}$

$$\sum_1^n K(\rho(x,x_i)/h_n) 1_{r_p(x)<\rho(x,x_i)\leq r_p^\varepsilon(x)} \leq 3n\varepsilon K(r_p(x)/h_n).$$

Finally, we have that

$$\sum_1^n w_i^S(x) \geq \sum_1^n K(r_p(x)/h_n) 1_{\rho(x,x_i)\leq r_p(x)} \geq n(p-\varepsilon)K(r_p(x)/h_n).$$

Therefore, using all three of our inequalities, we have that for any $x \in \mathscr{X}$

$$\begin{aligned}
\sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p(x)} &= \sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p^\varepsilon(x)} + \sum_1^n w_i^S(x) 1_{r_p^\varepsilon \geq \rho(x,x_i)>r_p(x)} \\
&= \frac{\sum_1^n K(\rho(x,x_i)/h_n) 1_{\rho(x,x_i)>r_p^\varepsilon(x)} + \sum_1^n K(\rho(x,x_i)/h_n) 1_{r_p^\varepsilon \geq \rho(x,x_i)>r_p(x)}}{\sum_1^n K(\rho(x,x_i)/h_n)} \\
&\leq \frac{n\delta K(r_p(x)/h_n)) + 3n\varepsilon K(r_p(x)/h_n)}{n(p-\varepsilon)K(r_p(x)/h_n).} \\
&= \frac{\delta + 3\varepsilon}{p-\varepsilon}.
\end{aligned}$$

If S does *not* satisfy equation B.5, then we simply have $\sup_{x\in\mathscr{X}} \sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p(x)} \leq 1$. Combining all of this, we have that

$$E_{S\sim\mathscr{D}^n} \sum_1^n w_i^S(x) 1_{\rho(x,x_i)>r_p(x)} \leq \delta(1) + (1-\delta)\frac{\delta+3\varepsilon}{p-\varepsilon}.$$

Since δ, ε can be made arbitrarily small, the result follows. □

By assumption, $\mathscr{X}$ is compact and therefore has diameter $D < \infty$. Define

$$t_n = \sqrt{n \log n K(\frac{D}{h_n})} \text{ for } 1 \leq n < \infty.$$

Lemma B.2.21. $\lim_{n\to\infty} E_{S\sim D^n}[t_n \sup_{x\in\mathscr{X}} w_i^S(x)] = 0.$

Proof. Because K is a decreasing function, we have that $K(D/h_n) \leq K(\rho(x,x_i)/h_n) \leq K(0)$. As

a result, we have that for any $x \in \mathscr{X}$,

$$\begin{aligned} t_n \sup_{1 \le i \le n} w_i^S(x) &= \frac{t_n \sup_{1 \le i \le n} K(\rho(x,x_i)/h_n)}{\sum_1^n K(\rho(x,x_i)/h_n)} \\ &\le \frac{t_n K(0)}{nK(D/h_n)} \\ &= K(0)\sqrt{\frac{n \log nK(D/h_n)}{n^2 K(D/h_n)^2}} \\ &= K(0)\sqrt{\frac{\log n}{nK(D/h_n)}}. \end{aligned}$$

However, by condition 4. of Corollary 2.4.6, $\lim_{n\to\infty} \frac{n}{\log n} K(D/h_n) = \infty$. Therefore, since the above inequality holds for all $x \in \mathscr{X}$, we have that

$$\lim_{n\to\infty} E_{S\sim D^n}[t_n \sup_{x \in \mathscr{X}} w_i^S(x)] \le \lim_{n\to\infty} K(0)\sqrt{\frac{\log n}{nK(D/h_n)}} = 0.$$

□

Lemma B.2.22. $\lim_{n\to\infty} E_{S\sim D^n} \frac{\log T(W,S)}{t_n} = 0.$

Proof. For $S \sim \mathscr{D}^n$, recall that $T(W,S)$ was defined as

$$T(W,S)|\{W_{x,\alpha,\beta} : x \in \mathscr{X}, 0 \le \alpha, 0 \le \beta \le 1\}|,$$

where $W_{x,\alpha,\beta}$ denotes

$$W_{x,\alpha,\beta} = \{i : \rho(x,x_i) \le \alpha, w_i^S(x) \ge \beta\}.$$

Our goal will to be upper bound $\log T(W,S)$.

The key observation is that $W_{x,\alpha,\beta}$ is precisely the set of x_i for which $\rho(x,x_i) \le r$ where r is some threshold. This is because the restriction that $w_i^S(x) \ge \beta$ can be directly translated into $\rho(x,x_i) \le r$ for some value of r, as K is a monotonically decreasing function. Thus, $T(W,S)$ is the number of subsets of S that can be obtained by considering the interior of some ball $B(x,r)$

centered at x with radius r.

We now observe that the set of closed balls in $\mathbb{R}^d$ has VC-dimension at most $d+2$. Thus by Sauer's lemma, there are at most $O(n^{d+2}$ subsets of $\{x_1, x_2, \ldots, x_n\}$ that can be obtained from closed balls. Thus $T(W,S) \leq O(n^{d+2}$.

Finally, we see that

$$\lim_{n\to\infty} \frac{\log T(W,S)}{t_n} = \lim_{n\to\infty} \frac{O(d\log n)}{\sqrt{n\log nK(\frac{D}{h_n})}} \leq \lim_{n\to\infty} \sqrt{\frac{O(d\log n)}{nK(\frac{D}{h_n})}} = 0,$$

with the last equality holding by condition 4. of Corollary 2.4.6. □

Finally, we note that Corollary 2.4.6 is an immediate consequences of Lemmas B.2.17, B.2.20, B.2.21, and B.2.22, as we can simply apply Theorem 2.4.4.

B.3 Useful Technical Definitions and Lemmas

Lemma B.3.1. *Let μ be a measure over $\mathbb{R}^d$, and let $\mathscr{A}$ denote a countable collections of measurable sets A_i such that $\mu(\bigcup_{A\in\mathscr{A}} A) < \infty$. Then for all $\varepsilon > 0$, there exists a finite subset of $\mathscr{A}$, $\{A_1, \ldots, A_m\}$ such that*

$$\mu(A_1 \cup A_2 \cup \cdots \cup A_m) > \mu(\bigcup_{A\in\mathscr{A}} A) - \varepsilon.$$

Proof. Follows directly from the definition of a measure. □

B.3.1 The support of a distribution

Let μ be a probability measure over $\mathbb{R}^d$.

Definition B.3.2. *The **support** of μ, $supp(\mu)$, is defined as all $x \in \mathbb{R}^d$ such that for all $r > 0$, $\mu(B(x,r)) > 0$.*

From this definition, we can show that $supp(\mu)$ is closed.

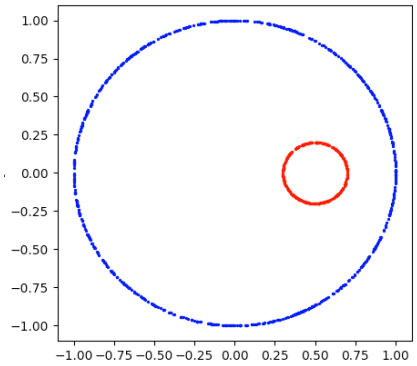


Figure B.1. Our data distribution $\mathscr{D} = (\mu, \eta)$ with μ^+ shown in blue and μ^- shown in red. Observe that this simple distribution captures varying distances between the red and blue regions, which necessitates having varying sizes for robustness regions.

Lemma B.3.3. *$supp(\mu)$ is closed.*

Proof. Let x be a point such that $B(x,r) \cap supp(\mu) \neq \emptyset$ for all $r > 0$. It suffices to show that $x \in supp(\mu)$, as this will imply closure.

Let x be such a point, and fix $r > 0$. Then there exists $x' \in B(x, r/2)$ such that $x' \in supp(\mu)$. By definition, we see that $\mu(B(x', r/3)) > 0$. However, $B(x', r/3) \subset B(x,r)$ by the triangle inequality. it follows that $\mu(B(x,r)) > 0$. Since r was arbitrary, it follows that $x \in supp(\mu)$. □

B.4 Experiment Details

Data Distribution

Our data distribution $\mathscr{D} = (\mu, \eta)$ is over $\mathbb{R}^2 \times \{\pm 1\}$, and is defined as follows. We let μ^+ consist of a uniform distribution over the circle $x^2 + y^2 = 1$, and μ^- consist of the uniform distribution over the circle $(x - 0.5)^2 + y^2 = 0.04$. The two distributions are weighted so that we draw a point from μ^+ with probability 0.7, and μ^- with probability 0.3. Finally, we utilize label noise 0.2 meaning that the label y matches that given by the Bayes optimal with probability 0.2. In summary, $\mathscr{D}$ can be described with the following 4 cases:

1. With probability 0.7×0.8, we select (x,y) with $x \in \mu^+$ and $y = +1$.

2. With probability 0.7×0.2, we select (x, y) with $x \in \mu^+$ and $y = -1$.

3. With probability 0.3×0.8, we select (x, y) with $x \in \mu^-$ and $y = -1$.

4. With probability 0.3×0.2, we select (x, y) with $x \in \mu^-$ and $y = +1$.

We also include a drawing (Figure B.1) of the support of $\mathscr{D}$, with the positive portion μ^+ shown in blue and the negative portion, μ^- shown in red.

Computing Robustness Regions

Recall that in order to measure robustness, we utilize the so-called partial neighborhood preserving regions V_x^κ (Definition 2.3.5) for varying values of κ. In the case of our data distribution $\mathscr{D}$, V_x^κ consists of points closer to x by a factor of κ than they are to μ^- (resp. μ^+) when $x \in \mu^+$ (resp. μ^-). To represent a region V_x^κ, we simply use a function f that verifies whether a given point $x' \in V_x^\kappa$. While this methodology is not sufficient for training general classifiers (for a whole litany of reasons: to begin with it assumes full knowledge of the distribution), it will suffice for our toy synthetic experiments.

Trained Classifiers

We train two classifiers, both of which are kernel classifiers.

The first classifier is an exponential kernel classifier with bandwidth function $h_n = \frac{1}{10\sqrt{\log n}}$ and kernel function $K(x) = e^{-x}$.

The second classifier is a polynomial kernel classifier with bandwidth function $h_n = \frac{1}{10n^{1/3}}$ and kernel function $K(x) = \frac{1}{1+x^2}$.

Both of these kernels are regular kernels, and both bandwidths satisfy sufficient conditions for consistency with respect to accuracy. In other words, both of these classifiers will converge towards the accuracy of the Bayes optimal.

However, the first classifier is selected to satisfy the criterion of Corollary 2.4.6, whereas the second is not. This distinction is reflected in our experiments.

Verifying Robustness

To verify the robustness of classifier f at point x (with respect to V_x^K), we simply do a grid search with grid parameter 0.01. We grid the entire regions into points with distance at most 0.01 between them, and then verify that f has the desired value at all of those points. To ensure proper robustness, we also simply verify that f cannot change enough within a distance of 0.01 by constructing an upper bound on how much f can possibly change. For kernel classifiers, this is simple to do as there is a relatively straightforward upper bound on the gradient of a Kernel classifier.

Appendix C

Appendix for Chapter 3

C.1 Expanded summary of [1]

In this section, we derive the formulation of Theorem 3.3.4 directly from their results. In particular, their results are not stated in terms of L_{rob} and L_{std}, and are instead framed in terms of different parameters. To account for this, we first review these alternative parameters, and then show how the statements in Theorem 3.3.4 can be

Recall, that [1] consider the setting in which the data distribution $\mathscr{D}_{\mu,\Sigma}$ can be characterized as a pair of Gaussians in $\mathbb{R}^d$, $\mathscr{N}(\mu,\Sigma)$ and $\mathscr{N}(-\mu,\Sigma)$, that are symmetric about the origin with each of them representing one label class. They consider robustness measured in any normed metric in $\mathbb{R}^d$, including the ℓ_p norm for $p \in [1,\infty]$.

For any such distribution (and robustness radius r), they introduce parameters $s_{rob}(\mu,\Sigma)$ and $s_{std}(\mu,\Sigma)$, which they refer to as the robust and standard signal-to-noise ratios respectively, that are defined as follows:

$$s_{std}(\mu,\Sigma) = 2\sqrt{\mu^t\Sigma^{-1}\mu},$$

$$s_{rob}(\mu,\Sigma) = \min_{||z||_p \leq r} 2\sqrt{(\mu-z)^t\Sigma^{-1}(\mu-z)},$$

where r represents the robustness radius and ℓ_p is the distance norm under which adversarial perturbations are measured.

They then show that these parameters fully characterize the sample complexity for robust and standard learning respectively. They express this through the following results:

1. Let Φ denote the cumulative density function of the standard normal distribution, and let $\overline{\Phi}(x) = 1 - \Phi(x)$. Then for any $\mathscr{D}_{\mu,\Sigma}$,

   - the optimally accurate classifier has standard loss $\overline{\Phi}(\frac{1}{2}s_{std})$.
   - the optimally robust classifier has robust loss $\overline{\Phi}(\frac{1}{2}s_{rob})$.

2. For any learning algorithm, there exists some mixture of $\mathscr{D}_{\mu,\Sigma}$ such that the expected robust loss is at least $\Omega(e^{(-\frac{1}{8}+o(1))s_{rob}^2}\frac{d}{n})$.

3. By contrast, for any distribution $\mathscr{D}_{\mu,\Sigma}$, it is possible to learn a classifier with expected standard loss at most $O(s_{std}e^{-\frac{1}{8}s_{std}^2}\frac{d}{n})$.

4. Thus, by (2.) and (3.), the gap between the robust sample complexity and the standard complexity can be bounded as

$$gap \geq \Omega\left(\frac{e^{(-\frac{1}{8}+o(1))s_{rob}^2}\frac{d}{n}}{s_{std}e^{-\frac{1}{8}s_{std}^2}\frac{d}{n}}\right) \simeq \Omega(e^{\frac{-1}{8}(s_{std}^2 - s_{rob}^2)}).$$

   They then qualitatively analyze this gap, and observe that for large values of μ and large values of r, this gap can be arbitrarily large, even as a function of d, the dimension.

We now show how to convert (2.), (3.), and (4.) into the statements appearing in Theorem 3.3.4. As before, let us define L_{std} and L_{rob} as the best possible standard and robust losses for $\mathscr{D}_{\mu,\Sigma}$ respectively. In particular, by (1.), we have

$$L_{std} = \overline{\Phi}(\frac{1}{2}s_{std}^2), \text{ and } L_{rob} = \overline{\Phi}(\frac{1}{2}s_{rob}^2).$$

We now express the bounds in (2.) and (4.) in terms of L_{std} and L_{rob}. To do so, we use the well

known inequality bounding $\overline{\Phi}(x)$ as

$$\Omega(\frac{x}{x^2+1}e^{-x^2/2}) < \Phi(x) < O(\frac{e^{-x^2/2}}{x}).$$

Substituting this into (2.) through (4.) imply the following, alternative forms.

2. For any learning algorithm, there exists some mixture of Gaussians, $\mathscr{D}_{\mu,\Sigma}$ such that the expected robust loss is at least $\Omega(L_{rob}\frac{d}{n})$.

3. For any distribution $\mathscr{D}_{\mu,\Sigma}$, it is possible to learn a classifier with expected standard loss at most $O(L_{std}\frac{d}{n})$.

4. By (2.) and (3.), the gap between robust sample complexity and standard sample complexity can be expressed as

$$gap \geq \Omega(\frac{L_{rob}}{L_{std}}).$$

Together, these three statements comprise Theorem 3.3.4.

C.1.1 The limiting case

While a core difference between our works is that we consider separated distributions whereas Gaussians are non-separated, we now consider the limiting case in which a pair of Gaussians *appear* separated. To do this, we will consider a case in which L_{rob} is small, and $n \sim O(\frac{1}{L_{rob}})$. In this case, with high probability, a sample of size n will *appear* linearly r-separated. Examining the bound in part 1 of Theorem 3.3.4, we see that their lower bound on the expected robust loss reduces to $O(\frac{1}{n}\frac{d}{n}) = O(\frac{d}{n^2})$, which is significantly weaker than ours (Theorem 3.3.2). Thus, considering Gaussians that appear linearly r-separated does not generalize to the general, linearly r-separated case.

C.2 Proof of Theorem 3.3.2

We begin by broadly outlining our proof of Theorem 3.3.2. Let Π be a probability distribution over $\mathscr{F}_{r,\rho}$, and let A be a learning algorithm that returns a linear classifier.

1. Sample $\mathscr{D} \sim \Pi$.
2. Sample $S \sim \mathscr{D}^n$.
3. Learn the classifier A_S using algorithm A and training sample S.
4. Evaluate A_S on $\mathscr{D}$. That is, compute $\mathscr{L}_r(A_S, \mathscr{D})$.

The basic idea of our proof is to show that for an appropriate choice of Π, the overall expected loss of this procedure, $\mathscr{L}_r(A_S, \mathscr{D})$, satisfies

$$\mathbb{E}_{D\sim\Pi}[\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})]] \geq \Omega(\frac{d}{n}).$$

Our primary method for doing this is switching expectations. In particular, observe that

$$\mathbb{E}_{D\sim\Pi}[\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})]] = \mathbb{E}_{S\sim\Sigma}[\mathbb{E}_{\mathscr{D}\sim\Pi|S}[\mathscr{L}_r(A_S, \mathscr{D})]],$$

where Σ denotes the distribution over all S obtained from first sampling $\mathscr{D} \sim \Pi$ and then sampling $S \sim \mathscr{D}^n$, and $\Pi|S$ denotes the posterior distribution of $\mathscr{D}$ after observing S. It then suffices to bound the quantity $\mathbb{E}_{\mathscr{D}\sim\Pi|S}[\mathscr{L}_r(A_S, \mathscr{D})]$, which is a significantly more tractable problem since we no longer need to deal with any specifics of the Algorithm A. In particular, S is fixed in this expectation and consequently A_S is just a fixed linear classifier. This bound subsequently follows from the distribution $\Pi|S$ having enough "variation" for this expectation to be sufficient large.

Our proof will have the following main steps, each of which is given its own subsection.

1. In section C.2.1, we construct the distribution Π, and prove several important properties about it.

2. In section C.2.2, we show that the desired property of Π holds, by bounding $\mathbb{E}_{\mathscr{D}\sim\Pi|S}[\mathscr{L}_r(A_S,\mathscr{D})]$. ∎

C.2.1 Constructing Π

We let r be a fixed robustness radius, and ℓ_p be our norm with which we measure robustness. Our construction of Π is a somewhat technical and lengthy process. We will organize this construction into 4 subsections, outlined here:

- In section C.2.1, we define the distribution $\mathscr{D}_a$, characterized by parameter $a \in [0,1]^d$. This forms the basis for constructing Π, which will comprise of distributions $\mathscr{D}_a$ for certain choices of a. We also show that $\mathscr{D}_a$ is linearly r-separated.
- In section C.2.1, we define the constant Δ, which will be essential for specifying which values of parameter a are permissible.
- In section C.2.1, we define functions $g_1, g_2 : [0, \frac{\Delta}{3}] \to [0, \frac{\Delta}{3}]$ that will be used to construct Π.
- In section C.2.1, we finally put together the previous 3 sections and construct Π. We also show that any $\mathscr{D}_a \sim \Pi$ satisfies $\rho(\mathscr{D}_a) \leq C$.

Defining $\mathscr{D}_a$

Let $e_1, e_2, \ldots, e_d$ denote the standard normal basis in $\mathbb{R}^d$. Define $v_i = Re_i$ and $u = \frac{R}{\sqrt{d}} \sum_1^d e_i$, where $R = \frac{9rd^{1/q}}{2\sqrt{d}}$. It will also be convenient to define the following function, which we will frequently use throughout the entirety of the appendix.

Definition C.2.1. *For $1 \leq l \leq \infty$, let $f_l : [0,1]^d \to \mathbb{R}^+$ be the function defined as*

$$f_l(a) = \sqrt[l]{\sum_1^d |\frac{1}{\sqrt{d}} + \overline{a} - a_i|^l},$$

where $\overline{a} = \frac{1}{d}\sum_1^d a_i$. For $l = \infty$, we take the convention that $\sqrt[\infty]{\sum_1^d |x_i|^\infty} = \max_{1\leq i\leq d} |x_i|$.

To define $\mathscr{D}_a$, we first define the concept of a line segment in $\mathbb{R}^d$.

Definition C.2.2. *Let $x_1, x_2 \in \mathbb{R}^d$ be two points. A **line segment** joining x_1, x_2 is defined as one of the following four sets.*

- $(x_1, x_2) = \{tx_1 + (1-t)x_t : 0 < t < 1\}$.
- $[x_1, x_2) = \{tx_1 + (1-t)x_t : 0 \leq t < 1\}$.
- $(x_1, x_2] = \{tx_1 + (1-t)x_t : 0 < t \leq 1\}$.
- $[x_1, x_2] = \{tx_1 + (1-t)x_t : 0 \leq t \leq 1\}$.

We will always distinguish which set we mean by using the notation above. In all cases, x_1, x_2 are said to be the endpoints of the line segment.

We now define $\mathscr{D}_a$.

Definition C.2.3. *Let $a \in [0,1]^d$ be a vector, and let $\overline{a} = \frac{1}{d}\sum_1^d a_i$. Set $\lambda_a = \frac{r}{R} f_q(a)$, where q is the dual norm of p. Assume that for all $1 \leq i \leq d$, $a_i > \lambda_a$ (i.e. we only $\mathscr{D}_a$ for a for which this holds). Let S^- and S^+ be two sets of d disjoint line segments (as defined in Definition C.2.2) defined as*

$$S^- = \{[v_i, v_i + (a_i - \lambda_a)u) : 1 \leq i \leq d\},$$

$$S^+ = \{(v_i + (a_i + \lambda_a)u, v_i + u] : 1 \leq i \leq d\}.$$

Then D_a is defined as the probability distribution of random variables (X,Y) where

- *X is chosen by the following random procedure. First, sample an arbitrary segment from $S^+ \cup S^-$ with each segment chosen with probability proportional to its ℓ_2 length. Next, X is selected from the uniform distribution over the chosen line segment. In particular, the probability that X lies on any interval on any line segment contained within $S^+ \cup S^-$ is directly proportional to the length of the interval.*

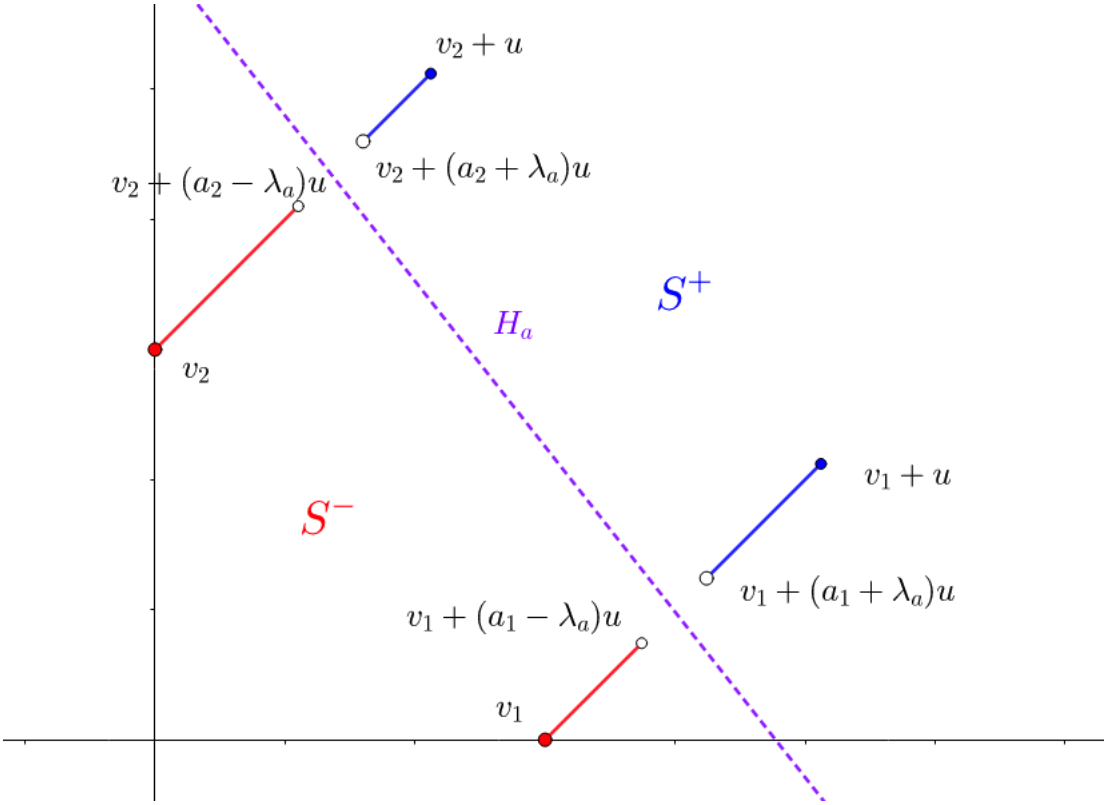


Figure C.1. An illustration of $\mathscr{D}_a$ in two dimensions. S^- is shown in red, and S^+ is shown in blue. The decision boundary, H_a, of the optimal linear classifier, $f_{w^a,1}$, is shown in purple.

- *Y is -1 if $X \in \cup S^-$ and $+1$ if $X \in \cup S^+$.*

We include an example of such a distribution in Figure C.1. Next, we explicitly compute a linear classifier that linearly r-separates $\mathscr{D}_a$.

Definition C.2.4. *Let $a \in [0,1]^d$, and let $\overline{a} = \sum_{i=1}^d a_i$. Then let w^a be defined as*

$$w_i^a = \frac{1}{R} - \frac{da_i}{R\sqrt{d} + dR\overline{a}}.$$

Lemma C.2.5. *w^a satisfies $\langle w^a, u \rangle = \frac{d}{\sqrt{d}+d\overline{a}}$ and $\langle w^a, v_i + a_i u \rangle = 1$, for all $1 \leq i \leq d$.*

Proof. By the definitions of v_i, u, we have that

$$\begin{aligned}
\langle w^a, u\rangle &= \langle w^a, \frac{1}{\sqrt{d}}\sum_1^d v_i\rangle \\
&= \frac{1}{\sqrt{d}}\sum_1^d Rw_i^a \\
&= \frac{1}{\sqrt{d}}\sum_1^d 1 - \frac{da_i}{\sqrt{d}+d\overline{a}} \\
&= \frac{1}{\sqrt{d}}\sum_1^d \frac{\sqrt{d}+d\overline{a}-da_i}{\sqrt{d}+d\overline{a}} \\
&= \frac{1}{\sqrt{d}}\frac{d\sqrt{d}}{\sqrt{d}+d\overline{a}} = \frac{d}{\sqrt{d}+d\overline{a}},
\end{aligned}$$

Which proves the first claim. Next, we also have that $\langle w^a, v_i\rangle = Rw_i^a$. Summing these, we get

$$Rw_i^a + \frac{da_i}{\sqrt{d}+d\overline{a}} = 1 - \frac{da_i}{\sqrt{d}+d\overline{a}} + \frac{da_i}{\sqrt{d}+d\overline{a}} = 1,$$

as desired. □

We now prove that $\mathscr{D}_a$ is linearly r-separated.

Lemma C.2.6. *$\mathscr{D}_a$ is linearly r-separated by the classifier $f_{w_a,1}$.*

Proof. Let H_a denote the hyperplane passing through $\{v_i + a_i u : 1 \le i \le d\}$. By Lemma C.2.5, H_a is the decision boundary of $f_{w_a,1}$. Referring to Figure C.1, we see that $\cup S^+$ lies entirely above H_a while the set $\cup S^-$ lies entirely below the hyperplane H_a, which the classifier $f_{w^a,1}$ has accuracy 1 with respect to $\mathscr{D}_a$. It suffices to show that $f_{w^a,1}$ is robust everywhere. In order to do this, we must show that all points in the support of $\mathscr{D}_a$ have ℓ_p distance at least r from H_a.

Fix any $1 \le i \le d$. Since the ℓ_p distance metric is invariant under translation and scales linearly with dilations, it follows that the point $x_i = v_i + (a_i - \lambda_a)u$ is the closest point on the segment $[v_i, v_i + (a_i - \lambda_a)u)$ to H_a. Suppose x_i has distance D under the ℓ_p norm to H_a. Then the key observation is that the ℓ_p ball, $B_p(x_i, D)$, must be tangent to H_a. Expressing this as an

equation, we have $\max_{z \in B_p(x_i,D)} \langle z, w^a \rangle = 1$, which can be re-written as

$$\max_{||z-x_i||_p \leq D} \langle z - x_i, w^a \rangle = 1 - \langle x_i, w^a \rangle.$$

By Lemma C.2.5 , $\langle w^a, u \rangle = \frac{d}{\sqrt{d}+d\overline{a}}$ and $\langle w^a, v_i + a_i u \rangle = 1$. Substituting this, we see that

$$\begin{aligned}
1 - \langle x_i, w^a \rangle &= 1 - \langle v_i + a_i u - \lambda_a u, w^a \rangle \\
&= 1 - \langle v_i + a_i u, w^a \rangle + \langle \lambda_a u, w^a \rangle \\
&= \langle \lambda_a u, w^a \rangle \\
&= \frac{d\lambda_a}{\sqrt{d}+d\overline{a}}.
\end{aligned}$$

However, by using the dual norm, we see that $\max_{||z-x_i||_p \leq D} \langle z - x_i, w^a \rangle = D||w^a||_q$. Thus it follows that

$$\begin{aligned}
D &= \frac{d\lambda_a}{(\sqrt{d}+d\overline{a})||w^a||_q} \\
&= \frac{d\frac{r}{R} f_q(a)}{(\sqrt{d}+d\overline{a})||w^a||_q} \\
&= \frac{d\frac{r}{R} \sqrt[q]{\sum_1^d |\frac{1}{\sqrt{d}} + \overline{a} - a_i|^q}}{(\sqrt{d}+d\overline{a})||w^a||_q} \\
&= \frac{r\sqrt[q]{\sum_1^d |\frac{1}{R}\frac{\sqrt{d}+d\overline{a}-da_i}{(\sqrt{d}+d\overline{a})}|^q}}{||w^a||_q} \\
&= \frac{r||w^a||_q}{||w^a||_q} = r.
\end{aligned}$$

We can use an analogous argument holds for $v_i + (a_i + r_a)u$, the closest point to H_a in S^+. Thus each point in the support of D^a has distance strictly larger than r (as the endpoints were not included) to H_a. Consequently $f_{w^a,1}$ linearly r-separates D^a, as desired. □

Defining Δ

Now that we have defined $\mathscr{D}_a$, we turn our attention to defining Π, which requires us to specify a distribution over valid choices of a. In particular, although $\mathscr{D}_a$ is defined for $a \in [0,1]^d$, we will require a more stringent condition on a for our construction to work. To this end, we begin by defining Δ, a key parameter that characterizes the domain of a. To define Δ, we use the following lemma.

Lemma C.2.7. *There exists a real number $\Delta > 0$ such that for all $l \in \{2, q\}$, and for all $a \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$,*

$$||\nabla f_l(a)||_2 \leq \frac{1}{d^2\sqrt{d}},$$

where f_l is as defined in Definition C.2.1.

Proof. Since $1 \leq q < \infty$, we see that for both choices of l, the function $h_l(x) = (\frac{1}{\sqrt{d}} - x)^l$ is a convex function for $x \in [-\frac{1}{2\sqrt{d}}, \frac{1}{2\sqrt{d}}]$. Thus, if $\sum_1^d x_i = 0$, then by Jensen's inequality, $\sum_1^d h_l(x_i) \geq \sum_1^d h_l(0)$. Applying this, we see that for all $l \in \{2, q\}$ and for all $a \in [\frac{1}{2} - \frac{1}{4\sqrt{d}}, \frac{1}{2} + \frac{1}{4\sqrt{d}}]^d$,

$$\begin{aligned}
f_l(a) &= \sqrt[l]{\sum_1^d |\frac{1}{\sqrt{d}} + \overline{a} - a_i|^l} \\
&= \sqrt[l]{\sum_1^d \left(\frac{1}{\sqrt{d}} + \overline{a} - a_i\right)^l} \\
&= \sqrt[l]{\sum_1^d h_l(a_i - \overline{a})} \\
&\geq \sqrt[l]{\sum_1^d h_l(0)} \\
&= f_l((\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})),
\end{aligned}$$

with the first equality holding since $\overline{a} - a_i < \frac{1}{\sqrt{d}}$ and the first inequality holding since $\sum_1^d a_i - \overline{a} = 0$.

Thus $f_l(a)$ must be locally minimized when $a = (\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$, and it follows that

$$||\nabla f_l(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})||_2 = 0, \text{ for } l = 2, q.$$

Now observe that the map $H(a) = \max_{l \in \{2,q\}} ||\nabla f_l(a)||_2$ is a continuous map as long as $|a_i - \overline{a}| < \frac{1}{\sqrt{d}}$ for all $1 \leq i \leq d$. Thus there exists an open neighborhood U about $(\frac{1}{2}, \frac{1}{2}, \ldots, \frac{1}{2})$ such that $H(a) \leq \frac{1}{d^2\sqrt{d}}$ for all $a \in U$. Taking Δ so that $[\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d \subseteq U$ suffices. □

Definition C.2.8. *Let Δ be any constant for which Lemma C.2.7 holds. In particular, Δ only depends on ℓ_p, the robustness norm, and d, the dimension.*

Defining g_1 and g_2

In this section, we define functions $g_1, g_2 : [0, \frac{\Delta}{3}] \to [0, \frac{\Delta}{3}]$ which we will use to specify Π. Before defining g_1 and g_2, we will first prove several technical lemmas.

Lemma C.2.9. *Let $I \subseteq \mathbb{R}$ be an interval, and $\Phi : I \to \mathbb{R}$ be a strictly convex function. For any $s \in \mathbb{R}$ and $t \geq 0$, let $\Phi_s(t) = \Phi(s-t) + \Phi(s+t)$. Then Φ_s is a strictly increasing function.*

Proof. Fix s, and let $0 \leq t_1 < t_2$. Then we see that by Jensen's inequality (for strictly convex functions),

$$\Phi(s+t_1) < \frac{(t_2 - t_1)\Phi(s+t_2)}{t_1 + t_2} + \frac{2t_1\Phi(s-t_1)}{t_1 + t_2},$$

and

$$\Phi(s-t_1) < \frac{(t_2 - t_1)\Phi(s-t_2)}{t_1 + t_2} + \frac{2t_1\Phi(s+t_1)}{t_1 + t_2}.$$

Summing these inequalities, we see that

$$\begin{aligned}\Phi_s(t_1) &= \Phi(s-t_1)+\Phi(s+t_1)\\ &< \frac{(t_2-t_1)\Phi(s+t_2)}{t_1+t_2}+\frac{2t_1\Phi(s-t_1)}{t_1+t_2}+\frac{(t_2-t_1)\Phi(s-t_2)}{t_1+t_2}+\frac{2t_1\Phi(s+t_1)}{t_1+t_2}\\ &= \frac{t_2-t_1}{t_1+t_2}(\Phi(s+t_2)+\Phi(s-t_2))+\frac{2t_1}{t_1+t_2}(\Phi(s-t_1)+\Phi(s+t_1))\\ &= \frac{t_2-t_1}{t_1+t_2}\Phi_s(t_2)+\frac{2t_1}{t_1+t_2}\Phi_s(t_1).\end{aligned}$$

Rearranging this yields $\Phi_s(t_1) < \Phi_s(t_2)$, as desired. □

Lemma C.2.10. *Let $I \subseteq \mathbb{R}$ be an interval, $\Phi : I \to \mathbb{R}$ be a strictly convex continuous function, and $x, y, z \in I$ be real numbers with $x < y < z$. Let $\varepsilon > 0$ be such that $x - \varepsilon \in I$ and $y + \varepsilon \leq z - \varepsilon$. Then there exist unique $\delta, \gamma > 0$ such that the following hold:*

$$\delta + \gamma = \varepsilon,$$

$$\Phi(x-\delta)+\Phi(y+\varepsilon)+\Phi(z-\gamma) = \Phi(x)+\Phi(y)+\Phi(z)$$

Proof. Fix any ε satisfying the desired conditions, and define $\Theta : [0, \varepsilon] \to \mathbb{R}$ as $\Theta(t) = \Phi(x - t)+\Phi(y+\varepsilon)+\Phi(z+t-\varepsilon)$. Then, utilizing the definition of Φ_s from Lemma C.2.9, we see that

$$\Theta(t) = \Phi_{\frac{x+z-\varepsilon}{2}}(\frac{z-x-\varepsilon}{2}+t)+\Phi(y+\varepsilon).$$

By Lemma C.2.9, it follows that Θ is strictly increasing in t, and since Φ is continuous, so is Θ. Next, we bound $\Theta(0)$ and $\Theta(\varepsilon)$ to put us in the configuration to apply the intermediate value

theorem. To bound $\Theta(0)$, we have

$$\begin{aligned}\Theta(0) &= \Phi(x)+\Phi(y+\varepsilon)+\Phi(z-\varepsilon)\\ &= \Phi(x)+\Phi_{\frac{y+z}{2}}(\frac{z-y}{2}-\varepsilon)\\ &< \Phi(x)+\Phi_{\frac{y+z}{2}}(\frac{z-y}{2})\\ &= \Phi(x)+\Phi(y)+\Phi(z),\end{aligned}$$

and to bound $\Theta(\varepsilon)$, we have

$$\begin{aligned}\Theta(\varepsilon) &= \Phi(x-\varepsilon)+\Phi(y+\varepsilon)+\Phi(z)\\ &= \Phi_{\frac{x+y}{2}}(\frac{y-x}{2}+\varepsilon)+\Phi(z)\\ &> \Phi_{\frac{x+y}{2}}(\frac{y-x}{2})+\Phi(z)\\ &= \Phi(x)+\Phi(y)+\Phi(z).\end{aligned}$$

Together, these equations imply $\Theta(0) < \Phi(x)+\Phi(y)+\Phi(z) < \Theta(\varepsilon)$. Since Θ is strictly increasing and continuous, there exists a unique $\delta \in [0,\varepsilon]$ such that $\Theta(\delta) = \Phi(x)+\Phi(y)+\Phi(z)$. Setting $\gamma = \varepsilon - \delta$, we see that

$$\Theta(\delta) = \Phi(x-\delta)+\Phi(y+\varepsilon)+\Phi(z-\gamma) = \Phi(x)+\Phi(y)+\Phi(z),$$

as desired.

□

Next, we define a function that will be useful for simplifying notation, both in this section and subsequent ones.

Definition C.2.11. *Let Δ be as in definition C.2.8. For $x,y,z \in [0,\frac{\Delta}{3}]$, let*

$$F(x,y,z) = \sqrt[q]{\left(\frac{1}{\sqrt{d}}-x\right)^q+\left(\frac{1}{\sqrt{d}}-\frac{2\Delta}{3}+y\right)^q+\left(\frac{1}{\sqrt{d}}+\frac{2\Delta}{3}+z\right)^q}.$$

We now define g_1, g_2.

Corollary C.2.12. *Let Δ be as in definition C.2.8. There exist 1-Lipshitz, monotonically non-decreasing functions $g_1, g_2 : [0, \frac{\Delta}{3}] \to [0, \frac{\Delta}{3}]$ such that for all $t \in [0, \frac{\Delta}{3}]$, $g_1(t) + g_2(t) = t$ and $F(t, g_1(t), g_2(t)) = F(0,0,0)$.*

Proof. We have two cases.

Case 1: $1 < q < \infty$**:**

Let $\Phi : [-\Delta, \Delta] \to \mathbb{R}$ be defined as $\Phi(x) = (\frac{1}{\sqrt{d}} - x)^q$. Since $q > 1$, and $\Delta < \frac{1}{\sqrt{d}}$, Φ is strictly convex. Observe that

$$F(x,y,z)^q = \Phi(x) + \Phi(2\frac{\Delta}{3} - y) + \Phi(-2\frac{\Delta}{3} - z).$$

Next, fix any $t \in [0, \frac{\Delta}{3}]$. Then observe that $-\frac{2\Delta}{3} \geq -\Delta$ and that $\frac{2\Delta}{3} - t \geq 0 + t$. This puts us in the configuration to apply Lemma C.2.10. In particular, there exist unique reals $\delta_t, \gamma_t > 0$ such that

$$\delta_t + \gamma_t = t,$$

$$\Phi(-\frac{2\Delta}{3} - \delta_t) + \Phi(t) + \Phi(\frac{2\Delta}{3} - \gamma_t) = \Phi(-\frac{2\Delta}{3}) + \Phi(0) + \Phi(\frac{2\Delta}{3}).$$

We now define $g_1, g_2 : [0, \frac{\Delta}{3}] \to [0, \frac{\Delta}{3}]$ as

$$g_1(t) = \gamma_t \text{ and } g_2(t) = \delta_t.$$

Then it is clear that $F(0,0,0) = F(t, g_1(t), g_2(t))$ and $g_1(t) + g_2(t)$ (by directly substituting into the equations above). All that remains is to show that g_1 and g_2 are 1-Lipschitz.

Fix any $0 \leq t_1 < t_2 \leq \frac{\Delta}{3}$, and let $t_2 - t_1 = \varepsilon$. The key idea is to apply Lemma C.2.10 to $-\frac{2\Delta}{3} - g_2(t_1) < t_1 < \frac{2\Delta}{3} - g_1(t_1)$ and ε. To do so, we first check the conditions of the lemma.

We have that

$$-\frac{2\Delta}{3} - g_2(t_1) - \varepsilon \geq -\frac{2\Delta}{3} - t_1 - \varepsilon = -\frac{2\Delta}{3} - t_2 \geq -\Delta,$$

and

$$\begin{aligned} t_1 + \varepsilon &= t_2 \\ &\leq \frac{\Delta}{3} \\ &\leq \frac{2\Delta}{3} - t_2 \\ &= \frac{2\Delta}{3} - t_1 - \varepsilon \\ &\leq \frac{2\Delta}{3} - g_1(t_1) - \varepsilon. \end{aligned}$$

Thus ε satisfies the necessary conditions for Lemma C.2.10. Since Φ is strictly convex, by Lemma C.2.10, there exist unique $\delta, \gamma > 0$ with $\delta + \gamma = \varepsilon$ such that

$$\Phi(-\frac{2\Delta}{3} - g_2(t_1) - \delta) + \Phi(t_1 + \varepsilon) + \Phi(\frac{2\Delta}{3} - g_1(t_1) - \gamma) = \Phi(-\frac{2\Delta}{3} - g_2(t_1)) + \Phi(t_1) + \Phi(\frac{2\Delta}{3} - g_1(t_1)).$$

However, by the definition of g_1, g_2, we see that both of these quantities are equal to $F(0,0,0)^q$. Moreover, again by the definition of g_1, g_2, we also have that $g_1(t_2)$ and $g_2(t_2)$ are the unique real numbers in $[0, \frac{\Delta}{3}$ that satisfy

$$\Phi(-\frac{2\Delta}{3} - g_2(t_2)) + \Phi(t_2) + \Phi(\frac{2\Delta}{3} + g_1(t_2)) = F(0,0,0)^q.$$

Thus, it follows that $g_2(t_2) = g_2(t_1) + \delta$ and $g_1(t_2) = g_1(t_1) + \gamma$. However, $t_2 - t_1 = \varepsilon$, and $\delta, \gamma < \varepsilon$ (since they sum to ε). Thus, we see that $|g_1(t_2) - g_1(t_1)| \leq |t_2 - t_1|$ and $|g_2(t_2) - g_2(t_1)| \leq |t_2 - t_1|$. Since t_1 and t_2 were arbitrary, it follows that g_1 and g_2 are both 1-Lipschitz, as desired.

Finally, since $\delta, \gamma > 0$, it follows that $g_2(t_2) > g_2(t_1)$ and $g_1(t_2) > g_1(t_1)$. Since t_1, t_2 were arbitrary, it follows that g_1, g_2 are monotonically non-decreasing.

Case 2: $q = 1$

In this case, since $\Delta < \frac{1}{\sqrt{d}}$ (Lemma C.2.7), we see that $F(x,y,z) = \frac{3}{\sqrt{d}} + y + z - x$. Setting $g_1(t) = g_2(t) = \frac{t}{2}$ suffices, and clearly satisfies the desired properties. □

Definition C.2.13. *Let Δ be as defined in Definition C.2.8. We let $g_1, g_2 : [0, \frac{\Delta}{3}] \to [0, \frac{\Delta}{3}]$ be defined as any function satisfying the conditions of Corollary C.2.12.*

Putting it all together: defining Π

We are now ready to define Π. For convenience, we assume d is a multiple of 3.

Definition C.2.14. *Let Δ, g_1, and g_2 be as defined in Definitions C.2.8 and C.2.13. Then Π is defined as the distribution of distributions $\mathscr{D}_a$ where a is a random vector constructed as follows. Let $t_1, t_2, \ldots t_{d/3}$ be drawn i.i.d from the uniform distribution over $[0, \frac{\Delta}{3}]$. Then for $1 \le i \le d/3$, we let*

- $a_i = \frac{1}{2} + t_i$.
- $a_{i+d/3} = \frac{1}{2} + 2\frac{\Delta}{3} - g_1(t_i)$.
- $a_{i+2d/3} = \frac{1}{2} - 2\frac{\Delta}{3} - g_2(t_i)$.

Together the variables $a_1, a_2, \ldots, a_d$ compose a. Thus a random distribution $\mathscr{D} \sim \Pi$ can be constructed by sampling a as above and setting $\mathscr{D} = \mathscr{D}_a$.

We now show that for all $\mathscr{D}_a \sim \Pi$, λ_a (Definition C.2.3) is constant.

Lemma C.2.15. *There exists a constant Λ such that for all $\mathscr{D}_a \sim \Pi$, $\lambda_a = \Lambda$.*

Proof. Let $\mathscr{D}_a \sim \Pi$ be arbitrary. By Lemma C.2.12, for all $1 \le i \le d$, $g_1(t_i) + g_2(t_i) = t_i$.

Substituting this, we see that

$$\begin{aligned}
\overline{a} &= \frac{1}{d}\sum_{1}^{d} a_i \\
&= \frac{1}{d}\sum_{1}^{d/3}(\frac{1}{2}+t_i)+(\frac{1}{2}+\frac{2\Delta}{3}-g_1(t_i))+(\frac{1}{2}-\frac{2\Delta}{3}-g_2(t_i)) \\
&= \frac{1}{d}\sum_{1}^{d/3}\frac{3}{2} \\
&= \frac{1}{2}.
\end{aligned}$$

Recall that $\lambda_a = \frac{r}{R} f_q(a) = \frac{r}{R}\sqrt[q]{\sum_1^d |\frac{1}{\sqrt{d}}+\overline{a}-a_i|^q}$. By substituting that $\overline{a}=\frac{1}{2}$ and expressing each a_i in terms of t_i, we see that

$$\begin{aligned}
\lambda_a &= \frac{r}{R}\sqrt[q]{\sum_{1}^{d}|\frac{1}{\sqrt{d}}+\overline{a}-a_i|^q} \\
&= \frac{r}{R}\sqrt[q]{\sum_{i=1}^{d/3}\left|\frac{1}{\sqrt{d}}+\frac{1}{2}-(\frac{1}{2}+t_i)\right|^q+\left|\frac{1}{\sqrt{d}}+\frac{1}{2}-\left(\frac{1}{2}+\frac{2\Delta}{3}-g_1(t_i)\right)\right|^q+\left|\frac{1}{\sqrt{d}}+\frac{1}{2}-\left(\frac{1}{2}-\frac{2\Delta}{3}-g_2(t_i)\right)\right|^q} \\
&= \frac{r}{R}\sqrt[q]{\sum_{1}^{d/3}\left|\frac{1}{\sqrt{d}}-t_i\right|^q+\left|\frac{1}{\sqrt{d}}+g_1(t_i)-\frac{2\Delta}{3}\right|^q+\left|\frac{1}{\sqrt{d}}+g_2(t_i)+\frac{2\Delta}{3}\right|^q} \\
&= \frac{r}{R}\sqrt[q]{\sum_{1}^{d/3}F(t_i,g_1(t_i),g_2(t_i))^q},
\end{aligned}$$

where F is defined as in Definition C.2.11. Next, by Corollary C.2.12, $F(t_i,g_1(t_i),g_2(t_i)) =$

$F(0,0,0)$ for all $1 \leq i \leq \frac{d}{3}$. Thus, if we set $\Lambda = \frac{r}{R}(\frac{d}{3})^{1/q}F(0,0,0)$, we have

$$\begin{aligned}
\lambda_a &= \frac{r}{R}\sqrt[q]{\sum_1^{d/3} F(t_i, g_1(t_i), g_2(t_i))^q} \\
&= \frac{r}{R}\sqrt[q]{\sum_1^{d/3} F(0,0,0)^q} \\
&= \frac{r}{R}\sqrt[q]{\frac{d}{3}F(0,0,0)^q} \\
&= \frac{r}{R}(\frac{d}{3})^{1/q}F(0,0,0) = \Lambda,
\end{aligned}$$

proving the claim. □

Definition C.2.16. *We define $\Lambda = \frac{r}{R}(\frac{d}{3})^{1/q}F(0,0,0)$, where F is defined as in Definition C.2.11.*

Next, we compute upper and lower bounds on Λ, both of which will be useful for subsequent lemmas.

Lemma C.2.17. $\frac{1}{9} < \Lambda < \frac{1}{3}$.

Proof. By definition, $\Lambda = \frac{d}{3}^{1/q}F(0,0,0)$. Substituting the definition of f, we see that $F(0,0,0) = \sqrt[q]{|\frac{1}{\sqrt{d}}|^q + |\frac{1}{\sqrt{d}} - \frac{2\Delta}{3}|^q + |\frac{1}{\sqrt{d}} + \frac{2\Delta}{3}|^q}$, and consequently,

$$3^{1/q}|\frac{1}{\sqrt{d}} - \frac{2\Delta}{3}| \leq F(0,0,0) \leq 3^{1/q}|\frac{1}{\sqrt{d}} + \frac{2\Delta}{3}|.$$

By definition, $\frac{2\Delta}{3} < \frac{1}{2\sqrt{d}}$. It follows that

$$\frac{r}{R}\frac{d^{1/q}}{2\sqrt{d}} < \Lambda < \frac{r}{R}\frac{3d^{1/q}}{2\sqrt{d}}.$$

Finally, since $\frac{r}{R} = \frac{2\sqrt{d}}{9d^{1/q}}$, substituting this yields $\frac{1}{9} < \Lambda < \frac{1}{3}$, as desired. □

Next, we show that for all $\mathscr{D}_a \in \Pi$, the aspect ratio (Definition 3.2.9), $\rho(\mathscr{D}_a)$, is bounded by a constant.

Lemma C.2.18. *For all $\mathscr{D}_a \in \Pi$, we have $\rho(\mathscr{D}_a) \leq 18\sqrt{3}$.*

Proof. We first bound the ℓ_2 margin, $\gamma(\mathscr{D}_a)$ (Definition 3.2.8). Recall that the margin, $\gamma(\mathscr{D}_a)$ is described as the largest possible ℓ_2 distance from the support of $\mathscr{D}_a$ to the decision boundary of a linear classifier. Thus, we can lower bound $\gamma(\mathscr{D}_a)$ by computing the distance from the support of $\mathscr{D}_a$ to H_a, the decision boundary of $f_{w^a,1}$ (Definition C.2.4).

By referring to Figure C.1 (in Section C.2.1), it becomes clear that the closest point (under the ℓ_2 margin) from S^- to H_a is the point $v_i + (a_i - \lambda_a)u$, for some value of i. Thus it suffices to compute the ℓ_2 distance from this point to the plane H_a.

Recall that by Lemma C.2.5, the point $v_i + a_i u$ satisfies $\langle w^a, v_i + a_i u\rangle = 1$, and consequently must lie on the hyperplane H_a. Let D denote the ℓ_2 distance from $v_i + (a_i - \lambda_a)u$ to H_a. Since w^a is the normal vector to H_a, it follows that

$$\begin{aligned}
D &= \langle v_i + a_i u - (v_i + (a_i - \lambda_a)u), \frac{w^a}{||w^a||_2}\rangle \\
&= \frac{\langle \lambda_a u, w^a\rangle}{||w^a||_2} \\
&\overset{(1)}{=} \frac{\langle \Lambda u, w^a\rangle}{||w^a||_2} \\
&\overset{(2)}{=} \frac{\Lambda \frac{d}{\sqrt{d}+d\overline{a}}}{||w^a||_2} \\
&\overset{(3)}{=} \frac{\Lambda \frac{d}{\sqrt{d}+d\overline{a}}}{\sqrt{\sum_1^d \left(\frac{\sqrt{d}+d\overline{a}-da_i}{R(\sqrt{d}+d\overline{a}}\right)^2}} \\
&= \frac{R\Lambda}{\sqrt{\sum_1^d (\frac{1}{\sqrt{d}} + \overline{a} - a_i)^2}} \\
&\overset{(4)}{=} \frac{R\Lambda}{f_2(a)}.
\end{aligned}$$

Here, (1) holds by Lemma C.2.15, (2) holds by Lemma C.2.5, (3) holds by Definition C.2.4, and (4) holds by Definition C.2.1.

Next, observe that since $\mathscr{D}_a \sim \Pi$, we must have $a \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$. Thus it follows

that $||a-(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2})||_2 \leq \Delta\sqrt{d}$. However, by applying Lemma C.2.7, we also see that f_2 is $\frac{1}{d^2\sqrt{d}}$-Lipschitz over $[\frac{1}{2}-\Delta,\frac{1}{2}+\Delta]^d$. Thus, it follows that

$$f_2(a) \leq f_2(\frac{1}{2},\frac{1}{2},\ldots,\frac{1}{2}) + \Delta\sqrt{d}\frac{1}{d^2\sqrt{d}} \leq 2,$$

with the latter inequality holding from the definition of Δ.

Substituting this and applying Lemma C.2.17, we see that

$$\gamma(\mathscr{D}_a) \geq \frac{R\Lambda}{2} \geq \frac{R}{18}.$$

Next, to bound the aspect ratio, $\rho(\mathscr{D}_a)$, we must also bound the ℓ_2 diameter of $\mathscr{D}_a$. However, the ℓ_s diameter of $\mathscr{D}_a$ is $R\sqrt{3}$, since it is the distance from v_i+u to v_j for $i \neq j$. Thus, it follows that

$$\rho(\mathscr{D}_a) = \frac{diam_2(\mathscr{D}_a)}{\gamma(\mathscr{D}_a)} \leq \frac{R\sqrt{3}}{R/18} = 18\sqrt{3},$$

as desired. □

Note that a tighter analysis (and selection of Δ) can give a smaller bound for $\rho(\mathscr{D}_a)$, but the most important fact is that $\rho(\mathscr{D}_a) = O(1)$.

C.2.2 Bounding the expected robust loss

In this section, we finally prove our lower bound, Theorem 3.3.2. This will require a few important steps, which we have separated into the following subsections.

- In section C.2.2, we give a useful lower bound for the loss $\mathscr{L}_r(f,\mathscr{D}_a)$ where f is an arbitrary linear classifier.
- In section C.2.2, we give an explicit computation for the posterior distribution $\Pi|S$ where $S \sim \mathscr{D}_a^n$ is the observed training sample.
- Finally, in section C.2.2, we present the proof of Theorem 3.3.2.

Bounding the loss $\mathscr{L}_r(f, \mathscr{D}_a)$

In this section, we find a lower bound on the loss $\mathscr{L}_r(f, \mathscr{D}_a)$ where f is a linear classifier. We begin by first restricting f to be in the set of classifiers

$$f \in \{f_{w^b,1} : b \in [0,1]^d\},$$

where w^b is as defined in Definition C.2.4. These are precisely the classifiers that have a decision boundary that passes through some point on every line segment in $\{[v_i, v_i + u] : 1 \leq i \leq d\}$. We are able to only consider these classifiers since all other linear classifiers clearly have a very high loss with respect to $\mathscr{D}_a$ as they necessarily misclassify at least half the points on the line segment $[v_i, v_i + u]$ for some value of i.

We now find an initial lower bound on $\mathscr{L}_r(f_{w^b,1}, \mathscr{D}_a)$.

Lemma C.2.19. *Fix any $\mathscr{D}_a \in \Pi$, and let $b \in [0,1]^d$ be arbitrary. Let w^b be the vector defined as in Definition C.2.4, and $\lambda_b = \frac{r}{R} f_q(b)$ where f is as defined in Definition C.2.1. Then*

$$\mathscr{L}_r(f_{w^b,1}, \mathscr{D}_a) \geq \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda}.$$

Proof. By Lemma C.2.6, $f_{w^b,1}$ precisely r-separates $\mathscr{D}_b$. This implies that for all $1 \leq i \leq d$,

$$f_{w^b,1}(x) = \begin{cases} 1 & x \in (v_i + (b_i + \lambda_b)u, v_i + u] \\ -1 & x \in [v_i, v_i + (b_i - \lambda_b)u) \\ \text{not robust} & x \in [v_i + (b_i - \lambda_b)u, v_i + (b_i + \lambda_b)u] \end{cases}.$$

Without loss of generality, suppose that $b_i \geq a_i$. The key observation is that for all $1 \leq i \leq d$, if $x \in [v_i + (a_i + \lambda_a)u, v_i + (b_i + \lambda_b)u]$, then $f_{w^b,1}(x) = -1$ for $f_{w^b,1}$ is not robust at x. In both cases, we see that $f_{w^b,1}$ is either inaccurate or not robust for all points in $[v_i + (a_i + \lambda_a)u, v_i + (b_i + \lambda_b)u]$.

This interval has ℓ_2 length at least $(|a_i - b_i| + (\lambda_b - \lambda_a))||u||_2$. Note that in the case that

$a_i \leq b_i$ we can get an identical expression. Thus, combining this for all i, we see that $f_{w^b,1}$ is either inaccurate or not robust for a total length of $[d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|]||u||_2$. Dividing by the total length of the support of $\mathscr{D}_a$, we find that

$$\begin{aligned}
\mathscr{L}_r(f_{w^b,1}, \mathscr{D}_a) &\geq \frac{[d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|]||u||_2}{\sum_1^d |||[v_i, v_i + (a_i - \lambda_a)u) + (v_i + (a_i + \lambda_a)u, v_i + u]||_2} \\
&= \frac{[d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|]||u||_2}{\sum_1^d ||u_2||(1 - 2\lambda_a)} \\
&= \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d(1 - 2\lambda_a)} \\
&= \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda},
\end{aligned}$$

with the last equality holding since by Lemma C.2.15, $\lambda_a = \Lambda$. □

Lemma C.2.20. *For all $\mathscr{D}_a \in \Pi$ and $b \in [0,1]^d$, $d(\lambda_a - \lambda_b) \leq \frac{1}{2}\sum_1^d |a_i - b_i|$.*

Proof. We have two cases.

Case 1:

$b \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$.

Observe that $\lambda_b = \frac{r}{R} f_q(b)$ and $\lambda_a = \frac{r}{R} f_q(a)$. By Lemma C.2.7, we see that f_q is $\frac{1}{d^2\sqrt{d}}$-Lipschitz over the domain $[\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$. It follows that

$$\begin{aligned}
\lambda_a - \lambda_b &= \frac{r}{R}(f_q(a) - f_q(b)) \\
&\leq \frac{r}{R}||a - b||_2 \frac{1}{d^2\sqrt{d}} \\
&= \frac{2\sqrt{d}}{9d^{1/q}}||a - b||_2 \frac{1}{d^2\sqrt{d}} \\
&< \frac{||a - b||_1}{2d},
\end{aligned}$$

with the last inequality following since the ℓ_2 norm is smaller than the ℓ_1 norm. Rearranging this gives the statement of the Lemma as desired.

Case 2:

$b \notin [\frac{1}{2}-\Delta, \frac{1}{2}+\Delta]^d$.

The main idea in this case will be to find $b' \in [\frac{1}{2}-\Delta, \frac{1}{2}+\Delta]^d$ such that $\lambda_b \geq \lambda_{b'}$ and such that $||b'-a||_1 \leq ||b-a||_1$. We will then apply Case 1 to get the desired result.

Without loss of generality, assume that $b_1 \geq b_2 \geq \cdots \geq b_d$, and that $b_1, b_2, \ldots b_k > \frac{1}{2}+\Delta$, $b_{k+1}, \ldots, b_l \in [\frac{1}{2}-\Delta, \frac{1}{2}+\Delta]$, and $b_{l+1}, \ldots b_d < \frac{1}{2}-\Delta$ for some values of k and l.

We will construct b' in four steps. In each of these steps, we will change the values of b_i such that neither $||a-b||_1$ nor λ_b are increased. At each step, we let b_i refer to its value at the end of the previous step.

Finally, for reference, recall that

$$\lambda_b = \frac{r}{R} f_q(b) = \frac{r}{R} \sqrt[q]{\sum_1^d |\frac{1}{\sqrt{d}} + \overline{b} - b_i|^q}.$$

Step 1:

We set

$$b_i \leftarrow \begin{cases} \frac{1}{k}\sum_{j=1}^k b_j & 1 \leq i \leq k \\ b_i & k+1 \leq i \leq l \\ \frac{1}{d-l}\sum_{j=l+1}^d b_j & l+1 \leq i \leq d \end{cases}.$$

Since $a \in [\frac{1}{2}-\Delta, \frac{1}{2}+\Delta]^d$, we see that these operations do not change $||a-b||_1$, as $\sum_1^k |b_i - a_i| = \sum_1^k b_i - a_i$ and $\sum_{l+1}^d |b_i - a_i| = \sum_1^k a_i - b_i$. Also, observe that this operation preserves $\overline{b}$, and consequently since the function $f(x) = |\frac{1}{\sqrt{d}} + \overline{b} - x|^q$ is convex, we see that by Jensen's inequality that λ_b is not increased by this operation.

Step 2:

Let $\beta = \sum_1^k (b_i - \frac{1}{2} - \Delta) - \sum_{l+1}^d (\frac{1}{2} - \Delta - b_i)$. Then we set

$$b_i \leftarrow \begin{cases} \begin{cases} \frac{1}{2} + \Delta + \frac{\beta}{k} & 1 \le i \le k \\ b_i & k+1 \le i \le l \\ \frac{1}{2} - \Delta & l+1 \le i \le d \end{cases} & \beta \ge 0 \\ \begin{cases} \frac{1}{2} + \Delta & 1 \le i \le k \\ b_i & k+1 \le i \le l \\ \frac{1}{2} - \Delta + \frac{\beta}{d-l} & l+1 \le i \le d \end{cases} & \beta < 0 \end{cases}.$$

Observe that this operation cannot increase $||a-b||_1$, since it doesn't increase $|a_i - b_i|$ for any value of i. Furthermore, this operation also does not change $\overline{b}$, and a similar convexity argument on the function $f(x) = |\frac{1}{\sqrt{d}} + \overline{b} - x|^q$ can show that this does not increase λ_b.

Finally, if $\beta = 0$, we set $b' = b$, since we have reached a state such that $b \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$.

Step 3a:

We run this step if $\beta > 0$. Let $\alpha = \frac{\sum_{k+1}^d (\frac{1}{2} + \Delta - b_i)}{\beta}$. We then set

$$b_i \leftarrow \begin{cases} \begin{cases} \frac{1}{2} + \Delta & 1 \le i \le k \\ (\frac{1}{2} + \Delta)(\frac{\alpha - 1}{\alpha}) + \frac{b_i}{\alpha} & k+1 \le i \le d \end{cases} & \alpha \ge 1 \\ \begin{cases} \frac{1}{2} + \Delta + \frac{\beta}{k}(1-\alpha) & 1 \le i \le k \\ \frac{1}{2} + \Delta & k+1 \le i \le d \end{cases} & \alpha < 1 \end{cases}.$$

In this step, we can similarly verify that $||a-b||_1$ does not increase (as $|a_i - b_i|$ is strictly reduced for $1 \le i \le k$ by an exact amount to offset the possible increases in $|a_i - b_i|$ for $k+1 \le i \le d$). We also see by the same convexity argument as usual that this operation reduces λ_b.

Step 3b:

We run this step if $\beta < 0$. Let $\alpha = \frac{\sum_{k+1}^{d}(b_i - \frac{1}{2} + \Delta)}{\beta}$. We then set

$$b_i \leftarrow \begin{cases} \begin{cases} (\frac{1}{2} - \Delta)(\frac{\alpha - 1}{\alpha}) + \frac{b_i}{\alpha} & 1 \leq i \leq l \\ \frac{1}{2} - \Delta & k+1 \leq i \leq d \end{cases} & \alpha \geq 1 \\ \begin{cases} \frac{1}{2} - \Delta & 1 \leq i \leq l \\ \frac{1}{2} - \Delta + \frac{\beta}{d-l}(1-\alpha) & l+1 \leq i \leq d \end{cases} & \alpha < 1 \end{cases}.$$

The justification for this step is analogous to 3a.

Step 4:

We only run this step if $\alpha < 1$. Observe that if $\alpha \geq 1$, then both Step 3a and Step 3b result with $b \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$, which we set as b'. Observe that in this case, either $b_i \geq a_i$ for all i, or $b_i \leq a_i$ for all i. Thus we simply set

$$b_i \leftarrow \overline{b}.$$

This operation does not change $||a - b||_1$, and it also reduces λ_b (by a convexity argument).

Step 5:

Finally, for all $1 \leq i \leq d\Delta$, we set $b_i = \frac{1}{2} - \Delta$ if $\overline{b} < \frac{1}{2} - \Delta$ and otherwise set $b_i = \frac{1}{2} - \Delta$ if $\overline{b} > \frac{1}{2} + \Delta$. In both cases, λ_b is not changed, and $||a - b||_1$ is strictly reduced. In this step, we finally set $b' = b$. Note that we do not always reach this step, as it was possible in any of the previous steps to reach some $b \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$, at which point we would have simply terminated.

Conclusion:

Through steps 1 through 5, we have found $b' \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$ such that $\lambda_{b'} \leq \lambda_b$ and $||a - b'||_1 \leq ||a - b||_1$. By applying Case 1 to b', we see that $d(\lambda_a - \lambda_{b'}) \leq \frac{1}{2}||a - b'||_1$. Thus,

we have that

$$\frac{1}{2}||a-b||_1 \geq \frac{1}{2}||a-b'||_1 \geq d(\lambda_a - \lambda_{b'}) \geq d(\lambda_a - \lambda_b),$$

which implies the result by the transitive property.

□

From the previous two lemmas, we immediately have the following:

Corollary C.2.21. *For all* $\mathscr{D}_a \in \Pi$ *and* $b \in [0,1]^d$,

$$\mathscr{L}_r(f_{w^b,1}, \mathscr{D}_a) \geq \frac{1}{2d}\sum_1^d |a_i - b_i|.$$

Proof. We have that

$$\begin{aligned}
\mathscr{L}_r(f_{w^b,1}, \mathscr{D}_a) &\overset{(a)}{\geq} \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda} \\
&\geq \frac{\sum_1^d |a_i - b_i| - d(\lambda_a - \lambda_b) +}{d} \\
&\overset{(b)}{\geq} \frac{\sum_1^d |a_i - b_i| - \frac{1}{2}\sum_1^d |a_i - b_i|}{d} \\
&= \frac{1}{2d}\sum_1^d |a_i - b_i|,
\end{aligned}$$

where (a) holds by Lemma C.2.19 and (b) holds by Lemma C.2.20. □

Computing the posterior distribution, $\Pi|S$

Recall that our ultimate goal is to show that

$$\mathbb{E}_{\mathscr{D}\sim\Pi}[\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S, \mathscr{D})]] \geq \Omega(\frac{d}{n}),$$

where A denotes any learning algorithm returning a linear classifier. The main idea for showing this is to "switch expectations" and realize that

$$\mathbb{E}_{\mathscr{D}\sim\Pi}[\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S,\mathscr{D})]] = \mathbb{E}_{S\sim\Sigma}[\mathbb{E}_{\mathscr{D}\sim\Pi|S}[\mathscr{L}_r(A_S,\mathscr{D})]],$$

where $\Pi|S$ denotes the posterior distribution over Π after observing S. In this section, we fully characterize the distribution $\Pi|S$, and prove several important properties about it.

Recall (Definition C.2.14) that $\mathscr{D}_a \sim \Pi$ is generated by first choosing $t_1, t_2, \ldots, t_{d/3} \sim \mathbb{U}[0, \frac{\Delta}{3}]$ i.i.d, and then letting $a = (a_1, a_2, \ldots, a_d)$ be a function of $t = (t_1, \ldots, t_{d/3})$. Thus, to compute the posterior $\Pi|S$, it suffices to focus on the posterior distribution of $t|S$ for any $1 \leq i \leq \frac{d}{3}$. We begin by first defining the likelihood of observing S given that it is generated from parameter t.

Definition C.2.22. *Let $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be any set of n points in $\mathbb{R}^d \times \{\pm 1\}$, and let $t \in [0, \frac{\Delta}{3}]^{d/3}$ be a vector. Let $a \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$ be defined as in Definition C.2.14. That is, let*

- $a_i = \frac{1}{2} + t_i$.
- $a_{i+d/3} = \frac{1}{2} + \frac{2\Delta}{3} - g_1(t_i)$.
- $a_{i+2d/3} = \frac{1}{2} - \frac{2\Delta}{3} - g_2(t_i)$.

Then we define $L(S|t)$ as the likelihood of observing the set S from $\mathscr{D}_a^n$. In particular, for any measurable region of points $R \subseteq (\mathbb{R}^d \times \{\pm 1\})^n$, we have that

$$\mathbb{P}_{S\sim\mathscr{D}_a^n}[S \in R] = \int_{x\in R} L(x|t)dx.$$

Lemma C.2.23. *Let $S \subset \mathbb{R}^d \times \{\pm 1\}$ be a set with n points. Then for all $t \in [0, \frac{\Delta}{3}]^{d/3}$,*

$$L(S|t) \in \left\{0, \left(\frac{1}{(d-2\Lambda)||u||_2}\right)^n\right\},$$

where Λ is as defined in Definition C.2.16 and $L(S|t)$ is as defined in Definition C.2.22.

Proof. Let $\mathscr{D}_a$ be an arbitrary distribution in Π. Observe that $\mathscr{D}_a$ is uniform over the set of all points in its support. Thus for every point in its support, we have that the likelihood $L(x|t)$ satisfies $L(x|t) = \frac{1}{(d-2\Lambda)||u||_2}$.

Taking the product of this over all points in S, we get the desired result. Note that if S contains some point not in the support of $\mathscr{D}_a$, then the likelihood becomes 0, since the likelihood of observing some point not in the support of $\mathscr{D}_a$ is 0. □

Definition C.2.24. *For any dataset S, let P_S denote the set of all "permissible" t, that is $t \in [0, \frac{\Delta}{3}]^d$ such that $L(S|t) \neq 0$. Formally,*

$$P_S = \{t : L(S|t) > 0\}.$$

We now fully characterize P_S when S is drawn from some $\mathscr{D} \sim \Pi$.

Lemma C.2.25. *Fix $n > 0$. For all $\mathscr{D} \sim \Pi$ and $S \sim \mathscr{D}^n$, there exist intervals (possibly open, closed, half open) $I_1^S, I_2^S, \ldots, I_{d/3}^S \subseteq [0, \frac{\Delta}{3}]$ such that $P_S = \prod_1^{d/3} I_i^S$.*

Proof. Let $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$. Since $S \sim \mathscr{D}^n$, we see that for $1 \leq j \leq n$, x_j must satisfy $x_j \in [v_i, v_i + u]$ for some $1 \leq j \leq d$. Using this, for $1 \leq i \leq d$ let

$$s_i^- = \underset{\{x_j : x_j \in [v_i, v_i+u], y_j = -1\}}{\arg\max} ||x_j - v_i||_2,$$

and

$$s_i^+ = \underset{\{x_j : x_j \in [v_i, v_i+u], y_j = +1\}}{\arg\max} ||x_j - (v_i + u)||_2.$$

s_i^- and s_i^+ can be thought of as the points from S on segment $[v_i, v_i + u]$ that are closest to each other and labeled as $-$ and $+$ respectively. As a default, if no such points exist, we set $s_i^- = v_i$ and $s_i^+ = v_i + u$.

Next, consider any $t \in [0, \frac{\Delta}{3}]^{d/3}$, let $a \in [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]^d$ be defined as in Definition C.2.14. That is, let

- $a_i = \frac{1}{2} + t_i$.
- $a_{i+d/3} = \frac{1}{2} + \frac{2\Delta}{3} - g_1(t_i)$.
- $a_{i+2d/3} = \frac{1}{2} - \frac{2\Delta}{3} - g_2(t_i)$.

The key idea of this lemma is that $t \in P_S$ (i.e. $L(S|t) > 0$) if and only if for all $1 \leq i \leq d$,

$$[v_i + (a_i - \Lambda)u, v_i + (a_i + \Lambda)u] \subseteq (s_i^-, s_i^+).$$

To see this, observe that if the claim above holds, then we must have that $s_i^- \in [v_i, v_i + (a_i - \Lambda)u)$ and $s_i^+ \in (v_i + (a_i + \Lambda)u, v_i + u]$, and it consequently follows that all points in S are elements of the support of $\mathscr{D}_a$ (Definition C.2.3), as all other points in S are "further" from the interval $[v_i + (a_i - \Lambda)u, v_i + (a_i + \Lambda)u]$ than the points s_i^+ and s_i^-. Conversely, if $L(S|t) > 0$, we must have that $S \subseteq supp(\mathscr{D}_a)$, which immediately translates to the statement above. Thus, it suffices to find all t such that this condition holds.

To do this, observe that the interval $[v_i + (a_i - \Lambda)u, v_i + (a_i + \Lambda)u]$ is a line segment of length $2\Lambda||u||_2$ that is centered at the point $v_i + a_i u$. Thus, in order for this to be a sub-segment of (s_i^-, s_i^+), we only need that a_i satisfy $v_i + a_i u \in (s_i^- + \Lambda u, s_i^- - \Lambda u)$. This condition is equivalent to the condition that $a_i \in J_i^S$ for some open interval $J_i^S \subseteq [0,1]$, where J_i^S is only dependent on s_i^-, s_i^+ and Λ (which is a constant). In summary, there exist interval $J_1^S, J_2^S, \ldots, J_d^S$ such that $t \in P_S$ if and only if $a_i \in J_i^S$ for $1 \leq i \leq d$.

Finally, note that for $1 \leq i \leq d/3$, $a_i, a_{i+d/3}, a_{i+2d/3}$ are all functions of t_i, and moreover these functions are 1-lipschitz, and monotonic. As a consequence, by taking the intersections of the pre-images of these functions, we find that this condition holds if and only if $t_i \in I_i^S$ where I_i^S is some interval that is a subset of $[0, \frac{\Delta}{3}]^{d/3}$. This proves the claim. □

Corollary C.2.26. *For any $S \sim \mathscr{D}$ where $\mathscr{D} \sim \Pi$, let I_i^S be defined as in Lemma C.2.25 for $1 \leq i \leq d/3$. Then the posterior distribution $t|S$ is equal to the uniform distribution over the set $\prod_{1 \leq i \leq d/3} I_i^S$, where t_i is sampled from I_i^S.*

Proof. First, recall that our prior on t is $\mathbb{U}([0,\frac{\Delta}{3}]^d)$, where $\mathbb{U}$ denotes the uniform distribution. By Lemma C.2.23, we see that for all $t \in P_S$, $L(S|t) = \left(\frac{1}{(d-2\Lambda)||u||_2}\right)^n$, and for all other t, $L(S|t) = 0$. Furthermore, by Lemma C.2.25, we see that $P_S = \prod_1^{1\le i\le d/3} I_i^S$. Thus, applying Bayes rules gives the desired result. □

We conclude this section by lower bounding the expected length of the interval I_i^S, denoted $\ell(I_i^S)$.

Lemma C.2.27. *For an interval $(c,d) \subset \mathbb{R}$, we let its length, denoted $\ell((c,d))$ be defined as $\ell((c,d)) = d-c$. Then for $1 \le k \le d/3$, the expected length (taken over $\mathscr{D}_a \sim \Pi$ and $S \sim \mathscr{D}_a^n$) of the interval I_k^S is at least $\Omega(\frac{d}{n})$. That is,*

$$\mathbb{E}_{\mathscr{D}_a\sim\Pi}\mathbb{E}_{S\sim\mathscr{D}_a^n}[\ell(I_k^S)]] \ge \Omega(\frac{d}{n}).$$

Proof. Fix any $\mathscr{D}_{a^*} \sim \Pi$, and let t^* denote the value of t used to generate a (as in Definition C.2.14). We will show that $\mathbb{E}_{S\sim\mathscr{D}_{a^*}^n}[\ell(I_k^S)]] \ge \Omega(\frac{d}{n})$, for all $1 \le k \le d/3$. We begin by explicitly computing the interval I_k^S.

Fix $1 \le k \le d/3$. Then $t_k* \in [0,\frac{\Delta}{3}]$. Assume that $t_k^* > 0$; we will handle the case $t_k^* = 0$ separately. Recall from the proof of Lemma C.2.25 that for $1 \le i \le d$, we defined

$$s_i^- = \underset{\{x_j : x_j \in [v_i, v_i+u], y_j = -1\}}{\arg\max} ||x_j - v_i||_2,$$

and

$$s_i^+ = \underset{\{x_j : x_j \in [v_i, v_i+u], y_j = +1\}}{\arg\max} ||x_j - (v_i+u)||_2.$$

for $1 \le i \le d$.

Next let $t \in [0,\frac{\Delta}{3}]^{d/3}$ be a vector, and let $a \in [\frac{1}{2}-\Delta, \frac{1}{2}+\Delta]^d$ be defined as $a_k = \frac{1}{2}+t_k$, $a_{k+d/3} = \frac{1}{2} + \frac{2\Delta}{3} - g_1(t_k)$ and $a_{k+2d/3} = \frac{1}{2} - \frac{2\Delta}{3} - g_2(t_k)$, for $1 \le k \le d/3$. Note that g_1, g_2 are the functions defined in Definition C.2.13.

As we argued in the proof of Lemma C.2.25, it then follows that $t_k \in I_k^S$ if and only if

$$[v_i + (a_i - \Lambda)u, v_i + (a_i + \Lambda)u] \subseteq (s_i^-, s_i^+),$$

for $i = k, k + d/3, k + 2d/3$. Finally, as we did in Lemma C.2.25, for each $1 \leq i \leq d$, we define intervals $J_i^S \subseteq [\frac{1}{2} - \Delta, \frac{1}{2} + \Delta]$ such that $a_i \in J_i^S$ if and only if $[v_i + (a_i - \Lambda)u, v_i + (a_i + \Lambda)u] \subseteq (s_i^-, s_i^+)$.

We now have the following three claims.

Claim 1:

Let $\alpha = \min\left(\frac{||s_k^- - (v_k + (a_k^* - \Lambda)u)||_2}{||u||_2}, t_k^*\right)$. If $t_k \in (t_k^* - \alpha, t_k^*]$, then

$$[v_k + (a_k - \Lambda)u, v_k + (a_k + \Lambda)u] \subseteq (s_k^-, s_k^+).$$

Proof: First, observe that since s_k^+ and s_k^- were sampled from $\mathscr{D}_{a^*}$, it follows that

$$[v_k + (a_k^* - \Lambda)u, v_k + (a_k^* + \Lambda)u] \subseteq (s_i^-, s_i^+).$$

Consider any $t_k \in [t_k^* - \alpha, t_k^*]$. Then substituting the definitions of a_k, a_k^* imply that $a_k \in [a_k^* - \alpha, a_k^*]$. Because of this, it follows that

$$\begin{aligned} ||(v_k + (a_k - \Lambda)u) - (v_k + (a_k^* - \Lambda)u)||_2 &= ||(a_k - a_k^*)u||_2 \\ &< \alpha ||u||_2 \\ &\leq ||s_k^- - (v_k + (a_k^* - \Lambda)u)||_2, \end{aligned}$$

which implies that $v_k + (a_k - \Lambda)u \in (s_i^-, v_k + (a_k^* - \Lambda)u]$. Furthermore, the fact that $a_k \leq a_k^*$ implies that $v_k + (a_k + \Lambda)u \in (v_k + (a_k - \Lambda)u, v_k + (a_k^* + \Lambda)u]$.

Together, these observations imply the desired result, as it follows that

$$[v_k + (a_k - \Lambda)u, v_k + (a_k + \Lambda)u] \subset (s_k^-, v_k + (a_k^* + \Lambda)u] \subset (s_k^-, s_k^+).$$

■

Claim 2:

Let $\beta = \min\left(\frac{||s_{k+d/3}^+ - (v_{k+d/3} + (a_{k+d/3}^* + \Lambda)u)||_2}{||u||_2}, g_1(t_k^*)\right)$. If $t_k \in (g_1^{-1}(g_1(t_k^*) - \beta), t_k^*]$, then

$$[v_{k+d/3} + (a_{k+d/3} - \Lambda)u, v_k + (a_{k+d/3} + \Lambda)u] \subseteq (s_{k+d/3}^-, s_{k+d/3}^+).$$

Proof: First, we observe that β is well defined since g_1 is a monotonic 1-Lipschitz function, and consequently has an inverse. Next, we also see that $0 \leq g_1(t_k^*) - g_1(t_k) \leq \beta$. Substituting the definitions of a_k^*, a_k, it follows that $0 \leq a_k - a_k^* \leq \beta$ (notice the order switch). At this point, we can apply the same argument as in Claim 1 to get the desired result. ■.

Claim 3:

Let $\tau = \min\left(\frac{||s_{k+2d/3}^+ - (v_{k+2d/3} + (a_{k+2d/3}^* + \Lambda)u)||_2}{||u||_2}, g_2(t_k^*)\right)$. If $t_k \in (g_2^{-1}(g_2(t_k^*) - \tau), t_k^*]$, then

$$[v_{k+2d/3} + (a_{k+2d/3} - \Lambda)u, v_{k+2d/3} + (a_{k+2d/3} + \Lambda)u] \subseteq (s_{k+2d/3}^-, s_{k+2d/3}^+).$$

Proof: Completely analogous to Claim 2. ■.

Combining these claims, we see that if $t_k \in (t_k^* - \alpha, t_k^*] \cap (g_1^{-1}(g_1(t_k^*) - \beta), t_k^*] \cap (g_2^{-1}(g_2(t_k^*) - \tau), t_k^*]$, then $t_k \in I_k^S$. Since these three intervals all have an endpoint in t_k^*, it follows that there is an interval with length η that is a subset of I_k^S, where

$$\eta = \min(\ell((t_k^* - \alpha, t_k^*])), \ell((g_1^{-1}(g_1(t_k^*) - \beta), t_k^*]), \ell((g_2^{-1}(g_2(t_k^*) - \tau), t_k^*])).$$

However, by substituting that g_1, g_2 are 1-Lipschitz, we see that $\ell((g_1^{-1}(g_1(t_k^*) - \beta), t_k^*]) \geq \beta$

and $\ell((g_2^{-1}(g_2(t_k^*)-\tau),t_k^*])) \geq \tau$. Thus, it follows that

$$\ell(I_k^S) \geq \eta \geq \min(\alpha,\beta,\tau).$$

Thus it suffices to show that $\mathbb{E}_{S\sim\mathscr{D}_{a^*}}[\min(\alpha,\beta,\tau)] \geq \Omega(\frac{d}{n})$.

To do this, observe that

- $\alpha||u||_2$ is the distance from the closest point labeled $-$ on the segment $[v_k, v_k+u]$ to the point $v_k+(a_k^*-\Lambda)u$
- $\beta||u||_2$ is the distance from the closest point labeled $+$ on the segment $[v_{k+d/3}, v_{k+d/3}+u]$ to the point $v_{k+d/3}+(\Lambda+a_{k+d/3}^*)u$
- $\tau||u||_2$is the distance from the closest point labeled $+$ on the segment $[v_{k+2d/3}, v_{k+2d/3}+u]$ to the point $v_{k+2d/3}+(\Lambda+a_{k+2d/3}^*)u$.

Finally, it is not difficult to see that for sufficiently large n, with high probability each of these distances will be $\Omega(\frac{d}{n})$. This is because with high probability there will be $\Theta(\frac{n}{d})$ points on each of the respective line segments, and we are considering the closest point among them to some reference point. Thus, it follows that with high probability $\mathbb{E}_{S\sim\mathscr{D}_{a^*}}[\min(\alpha,\beta,tau)] \geq \Omega(\frac{d}{n})$, as desired. □

Putting it all together, the proof

We prove the following key lemma, which directly implies Theorem 3.3.2.

Lemma C.2.28. *Let M be any learning algorithm that outputs a linear classifier. For any training sample of points $S=\{(x_1,y_1),(x_2,y_2),\ldots,(x_n,y_n)\}$, we let M_S denote the classifier learned by M from $S\sim\mathscr{D}$. Then it follows that*

$$\mathbb{E}_{\mathscr{D}\sim\Pi}\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(M_S,\mathscr{D})]] \geq \Omega(\frac{d}{n}).$$

Proof. Let $\mathscr{F}_n$ denote the distribution over $(\mathbb{R}^d \times \{\pm 1\})^n$ defined as the composition $\mathscr{D} \sim \Pi$ and $S \sim \mathscr{D}^n$. That is, $S \sim \mathscr{F}_n$ follows the same distribution as $\mathscr{D} \sim \Pi, S \sim \mathscr{D}^n$. Then we can write the expectation above as

$$\mathbb{E}_{\mathscr{D}\sim\Pi}\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S,\mathscr{D})]] = \mathbb{E}_{S\sim\mathscr{F}_n}\mathbb{E}_{\mathscr{D}\sim(\Pi|S)}[\mathscr{L}_r(M_S,\mathscr{D})]],$$

where $\Pi|S$ denotes the posterior distribution of $\mathscr{D}$ conditioned on observing S. First, fix any such S. We will bound $\mathbb{E}_{\mathscr{D}\sim(\Pi|S)}[\mathscr{L}_r(M_S,\mathscr{D})]$. First, by reparametrizing in terms of $t \in [0,\frac{\Delta}{3}]^{d/3}$ and applying Corollary C.2.26, we have that

$$\mathbb{E}_{D\sim(\Pi|S)}[\mathscr{L}_r(M_S,\mathscr{D})] = \mathbb{E}_{t_1\sim\mathbb{U}(I_1^S)}[\dots[\mathbb{E}_{t_n\sim\mathbb{U}(I_{d/3})}[\mathscr{L}_r(M_S,\mathscr{D}_a)]\dots],$$

where $I_1^S, I_2^S, \dots, I_{d/3}^S \subset [0,\frac{\Delta}{3}]$ are the intervals defined in Lemma C.2.25, and a is defined as in Definition C.2.14.

Next, let $b \in [0,1]^d$ be such that $M_S = f_{w^b,1}$, where w^b is defined as in Definition C.2.4. Then it follows from Corollary C.2.21 that

$$\begin{aligned}\mathscr{L}_r(M_S,\mathscr{D}_a)] &\geq \frac{1}{20d}\sum_1^d |a_i - b_i| \\ &\geq \frac{1}{20d}\sum_1^{d/3} |\frac{1}{2} + t_i - b_i|\end{aligned}$$

with the last inequality coming from substituting the definition of a_i and (and ignoring a_i for $i > d/3$). We now take the expectation of this inequality over $t_1, t_2, \dots, t_{d/3}$. To do so, observe that by simple algebra, $\mathbb{E}_{t_i\sim\mathbb{U}(I_i^S)}|\frac{1}{2} + t_i - b_i| \geq \frac{\ell(I_i^S)}{4}$. Substituting this, we see that

$$E_{t_1\sim\mathbb{U}(I_1^S)}[\dots[\mathbb{E}_{t_n\sim\mathbb{U}(I_{d/3}^S)}[\mathscr{L}_r(M_S,\mathscr{D}_a)]\dots] \geq \frac{1}{80d}\sum_{i=1}^{d/3}\ell(I_i^S).$$

Finally, by taking expectations over $S \sim \mathscr{F}_n$, we see that

$$\begin{aligned}
\mathbb{E}_{\mathscr{D}\sim\Pi}\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(A_S,\mathscr{D})]] &= \mathbb{E}_{S\sim\mathscr{F}_n}\mathbb{E}_{\mathscr{D}\sim(\Pi|S)}[\mathscr{L}_r(M_S,\mathscr{D})]] \\
&\geq \mathbb{E}_{S\sim\mathscr{F}_n}\frac{1}{80d}\sum_{i=1}^{d/3}\ell(I_i^S) \\
&= \frac{1}{80d}\sum_{1}^{d/3}\mathbb{E}_{S\sim\mathscr{F}}[\ell(I_i^S)] \\
&= \frac{1}{80d}\sum_{1}^{d/3}\mathbb{E}_{\mathscr{D}\sim\Pi}\mathbb{E}_{S\sim\mathscr{D}^n}[\ell(I_i^S)] \\
&\geq \frac{1}{80d}\sum_{1}^{d/3}\Omega(\frac{d}{n}) = \Omega(\frac{d}{n}),
\end{aligned}$$

where the last step follows from Lemma C.2.27. □

Finally, we can prove Theorem 3.3.2.

Proof. (Theorem 3.3.2). First, by Lemmas C.2.6 and C.2.18, we see that $\Pi \subseteq \mathscr{F}_{r,\rho}$ (provided $\rho > 10$). Next, by Lemma C.2.28, for any n there must exists some $\mathscr{D} \sim \Pi$ such that $\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(M_S,\mathscr{D})] \geq \Omega(\frac{d}{n})$. Thus selecting this distribution suffices. This concludes the proof. □

C.3 Proofs for Algorithm 2

This section is divided into 2 parts. In section C.3.1, we show that for the case in which our data distribution $\mathscr{D}$ is linearly r-separated by some hyperplane through the origin, the desired error bound holds. That is, we prove Theorem 3.4.3 under this assumption.

Next, in section C.3.2, we show how to generalize Algorithm 2 to arbitrary linearly r-separated distributions, and subsequently prove Theorem 3.4.3 in the general case.

C.3.1 Origin Case

We begin by precisely stating the conditions required in the "origin" case. We assume the following properties hold for our data distribution $\mathscr{D}$. We let S_r^+ and S_r^- be defined as in section 3.4.

1. There exists $R > 0$ such that for all $x \in S_r^+ \cup S_r^-$, $||x||_2 \leq R$.

2. There exists a unit vector $u \in \mathbb{R}^d$ and $\gamma_r > 0$ such that

   - $\mathscr{L}_r(f_{u,0}, \mathscr{D}) = 0$, where $f_{u,0}$ denotes the linear classifier with decision boundary $\langle u, x \rangle = 0$.
   - $S_r^+ \cup S_r^-$ has distance at least γ_r from the decision boundary of f_w. That is, $||S_r^+ \cup S_r^- - H_{u,0}||_2 \geq \gamma_r$.

3. By the previous conditions, it follows that $\langle u, yx' \rangle \geq \gamma_r$ for all $(x,y) \sim \mathscr{D}$, and $x' \in B_p(x,r)$. This is because u is a unit vector.

Next, before analyzing Algorithm 2, we will first give a slight modification of the algorithm that lends itself to better analysis. The only difference is that in this new algorithm, we first randomly sample $k \sim \{1, 2, \ldots, n\}$, and then only train on the first r data-points of our

training sample.

Algorithm 8: Modified-Adversarial-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$,
$w \leftarrow 0$
$k \sim \mathbb{U}(\{0, 1, 2, \ldots, n\})$
for $i = 1 \ldots k$ **do**
    $z = \arg\min_{||z - x_i||_p \leq r} y_i \langle w, z \rangle$ **if** $\langle w, y_i z \rangle \leq 0$ **then**
        $w \leftarrow w + y_i z$
    end if
end for
return $f_{w,0}$

We will show that Algorithm 8 satisfies the guarantees of Theorem C.3.2. We begin with the following, key lemma.

Lemma C.3.1. *Under the assumptions above about $\mathscr{D}$, Algorithm 8 makes at most $\frac{R^2}{\gamma_r^2}$ updates to w.*

Proof. Let w_t denote our weight vector after we make t updates. Observe that $w_t = w_{t-1} + y_t x_t + z'$ where (x_t, y_t) denotes the point we made a mistake on, and $z' = \arg\min_{|z|_p \leq r} \langle w, z \rangle$. Letting $x'_t = x_t + y_t z'$, we see that $w_t = w_{t-1} + y_t x'_t$. Now the key observation is that $(x'_t, y_t) \in S_r^+ \cup S_r^-$, and as a result, it follows that $\langle u, y_t x'_t \rangle \geq \gamma_r$. Using this, we see that

$$\begin{aligned}
\langle u, w_t \rangle &= \langle u, w_{t-1} + y_t x'_t \rangle \\
&= \langle u, w_{t-1} \rangle + \langle u, y_t x'_t \rangle \\
&\geq \langle u, w_{t-1} \rangle + \gamma_r.
\end{aligned}$$

Thus, by a simple proof by induction, we see that $\langle w_t, u \rangle \geq t\gamma_r$.

Next, observe that we must have $\langle w_{t-1}, y_t x'_t \rangle \leq 0$. This is because w_{t-1} must missclassify (x'_t, y_t) (thus failing to be astute at (x_t, y_t)) in order for it to be updated. Substituting this, we see

that

$$\begin{aligned}
||w_t||_2 &= \sqrt{\langle w_t, w_t \rangle} \\
&= \sqrt{\langle w_{t-1} + x_t' y_t, w_{t-1} + x_t' y_\rangle} \\
&= \sqrt{\langle w_{t-1}, w_{t-1} \rangle + 2\langle w_{t-1}, x_t' y_t \rangle + \langle x_t', x_t' \rangle} \\
&\leq \sqrt{||w_{t-1}||_2^2 + 0 + R^2},
\end{aligned}$$

with the last inequality holding since $|x_t'|_2 \leq R$. Thus, by a simple proof by induction, we see that $||w_t||_2 \leq R\sqrt{t}$.

Finally, since u is a unit vector, it follows that $||w_t||_2 \geq \langle w_t, u$. Substituting our inequalities, we find that $R\sqrt{t} \geq \gamma_r t$ which implies that $t \leq \frac{R^2}{\gamma_r^2}$. Since t is the number of mistakes we make, the result follows. □

Lemma C.3.2. *Let $\mathscr{D}$ be a distribution with the assumptions above. For any $S \sim \mathscr{D}^n$, let f_S denote the classifier learned by Algorithm 8. Then*

$$\mathbb{E}_{S \sim \mathscr{D}^n} \mathscr{L}_r(f_S, \mathscr{D}) \leq \frac{R^2}{\gamma_r^2 (n+1)}.$$

This Theorem directly follows from the classic online to offline result (Theorem 3 of [42]). For completeness, we include a proof in our context.

Proof. Fix any n and consider running Algorithm 8 on $S \sim \mathscr{D}^n$. Let L_t denote the expected robust loss of our classifier conditioning on $k = t$, and let L^* denote the expected overall loss of our classifier. It follows that

$$\mathbb{E}_{S \sim \mathscr{D}^n} L^* = \frac{1}{n+1} \sum_{t=0}^{n} \mathbb{E}_{S \sim \mathscr{D}^n}[L^* | k = t] = \frac{1}{n+1} \sum_{t=0}^{n} \mathbb{E}_{S \sim \mathscr{D}^n}[L_t].$$

Next, let $T \sim \mathscr{D}^{n+1}$ be a separate i.i.d drawn sample, and suppose we run the adversarial perceptron algorithm on the entirety of T (i.e. rung Algorithm 8 on T by setting $k = n+1$). For

$1 \leq t \leq n+1$, let X_t be the indicator variable for whether the tth point in T requires an update on w (i.e. the classifier is not astute at w). There are two important observations to make.

First, we have that $\mathbb{E}_{T\sim\mathscr{D}^{n+1}}[X_t] = \mathbb{E}_{S\sim\mathscr{D}^n}[L_{t-1}]$. This is because X_t is an indicator variable for a classifier trained on precisely $t-1$ i.i.d training examples lacking astuteness for a randomly drawn point from $\mathscr{D}$. Second, we have that $\sum_{t=1}^{n+1} X_t \leq \frac{R^2}{\gamma_r^2}$. This is because each $\sum X_t$ is precisely the number of updates that perceptron makes on T, which is bounded by Lemma C.3.1. By combining these two observations, we see that

$$\begin{aligned}
\mathbb{E}_{S\sim\mathscr{D}^n}[L^*] &= \frac{1}{n+1}\sum_{t=0}^{n}\mathbb{E}_{S\sim\mathscr{D}^n}[L_t] \\
&= \frac{1}{n+1}\sum_{t=0}^{n}\mathbb{E}_{T\sim\mathscr{D}^{n+1}}[X_{t+1}] \\
&= \frac{1}{n+1}\mathbb{E}_{T\sim\mathscr{D}^{n+1}}[\sum_{t=1}^{n+1} X_t] \\
&\leq \frac{R^2}{\gamma_r^2(n+1)},
\end{aligned}$$

as desired. □

C.3.2 General Case

In general case, we no longer assume that the optimal classifier $f_{u,b}$ passes through the origin. To account for this, we will need to first adapt our algorithm. The basic idea is to simply append a 1 to the vectors x and increase the dimension d by 1. We are then left with solving a $d+1$ dimensional problem in which the data is once-again separated by a hyperplane passing through the origin.

We begin with two useful sets of notation.

Definition C.3.3. *We use the following notation:*

- *For any $x \in \mathbb{R}^d$ and $R \in \mathbb{R}$, we let $x|R \in \mathbb{R}^{d+1}$ denote the $d+1$ dimensional vector obtained by appending the value R to x.*

- *For $w \in \mathbb{R}^{d+1}$, let $||w||_q^*$ denote the ℓ_q norm of the first d coordinates of w.*
- *For $x \in \mathbb{R}^{d+1}$, let $B_p^*(x,r)$ denote all $z \in \mathbb{R}^{d+1}$ such that $||z-x||_p \leq r$ and such that z and x both share the same last coordinate.*
- *For $S = \{(x_1,y_1),\ldots,(x_n,y_n)\} \subset \mathbb{R}^{d+1} \times \{\pm 1\}$, let R_S denote $\max_{i \neq j} ||x_i - x_j||_2$.*

We now propose the following modified version of Algorithm 2, that is capable of handling any dataset, including ones that aren't separated by a hyperplane through the origin.

Algorithm 9: General-Adversarial-Perceptron

Input: $S = \{(x_1,y_1),\ldots,(x_n,y_n)\} \sim \mathscr{D}^n$,
$x_i' \leftarrow x_i - x_1$.
$R_S = diam_2(S)$
$w \leftarrow 0 \in \mathbb{R}^{d+1}$
Randomly permute S
Randomly choose $k \in \{1,2,3,\ldots,n\}$.
for $t = 1 \ldots k$ **do**
    if $\langle w, y_t(x_t|R_S)\rangle \leq r||w||_q^*$ **then**
        $z' = \arg\min_{|z|_p \leq r} \langle w, z|0\rangle$
        $w \leftarrow w + y_t(x_t|R_S) + z'|0$
    end if
end for
$w^* \leftarrow$ first d coordinates of w
$b \leftarrow$ the last element of w
Return $f_{w^*,\langle w^*,x_1\rangle - bR_S}$

The basic idea of the algorithm is to first translate S so that one point is the origin, and then append R_S to every vector in S so that each vector is now $d+1$ dimensional. After doing this, we apply Algorithm 2 as before with one important difference: for our adversarial attacks, we make sure to not change the last coordinate.

We now show that this algorithm has a similar performance to our old algorithm. We first prove a helpful lemma.

Lemma C.3.4. *Let $\mathscr{D}$ be any linearly r-separated distribution, and let $S \sim \mathscr{D}^n$ such that S has positively and negatively labeled examples. Let $x_i' = x_i - x_1$ for $1 \leq i \leq n$. Then the following hold.*

- *There exists a unit vector $u \in \mathbb{R}^{d+1}$ such that for all $(x_i, y_i) \in S$, $\min_{z \in B_p^*(x_i')} \langle u, y_i(z|R_S) \rangle \geq \frac{\gamma_r(\mathscr{D})}{\sqrt{2}}$.*
- *For all $(x_i, y_i) \in S$, $||x_i'|R_S||_2 \leq \sqrt{2} diam_2(\mathscr{D})$.*

Proof. Without loss of generality, we will assume $x_1 = 0$ so that we can safely ignore the differences between x_i' and x_i. Since $\mathscr{D}$ is r-separated, there exist w, b (with w a unit vector) such that

$$\langle w, zy \rangle \geq by + \gamma_r(\mathscr{D}),$$

for all $(x,y) \sim \mathscr{D}$ and $z \in B_p(x,r)$. Furthermore, since $x_1 = 0$, it follows that $||x||_2 \leq \text{diam}_2(\mathscr{D})$ for all $(x,y) \sim \mathscr{D}$. This immediately implies that $||x_i|R_S||_2 \leq \sqrt{\text{diam}_2(\mathscr{D})^2 + R_S^2} \leq \sqrt{2}\text{diam}_2(\mathscr{D})$, yielding the second part of the lemma.

For the first part, observe that we can rearrange the equation above, we see that

$$\langle w| - \frac{b}{R_S}, zy|R_S \rangle \geq \gamma_r(\mathscr{D}).$$

The key observation is that the first equation implies that $b \leq R_S$. This is because S contains positively and negatively labeled examples, and consequently $\langle w, x_i \rangle \geq b + \gamma_r(\mathscr{D}) > b$ for some x_i such that $|x_i| = R_S$. Thus, it follows that the unit vector $u = \frac{w|\frac{-b}{R_S}}{\sqrt{1+b^2/R_S^2}}$ has the desired property, by observing that $\sqrt{1 + b^2/R_S^2} \leq \sqrt{2}$. □

Lemma C.3.4 allows us to analyze the performance of Algorithm 9. The basic idea is that our performance on the transformed data in $\mathbb{R}^{d+1}$ is isomorphic to its performance on the

data in $\mathbb{R}^d$. As a consequence, we can apply the same argument as in Theorem C.3.2 to get a bound on the error estimate. However, this bound must be given in terms of the diameter and robust margin of the *transformed data*: quantities that have been bounded in Lemma C.3.4. Thus, putting this all together, Theorem 3.4.3 follows.

C.4 Details for Kernel Algorithm

Next, we find analogs of linear r-separability and the robust margin when considering kernels. First, we define an embedding function.

Definition C.4.1. *Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ be a kernel similarity function. Then there exists a Hilbert space H and map $\phi : \mathbb{R}^d \to H$ such that for all $x_1, x_2 \in \mathbb{R}^d$, we have*

$$K(x_1, x_2) = \langle \phi(x_1), \phi(x_2) \rangle.$$

*We call ϕ the **embedding function** and H the **embedding space**.*

The key idea of this section is that Kenrel classifiers correspond to linear classifiers in embedded space. This is the essence of the "kernel trick." Formally, we have the following, well-known theorem.

Theorem C.4.2. *Let $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^+$ be a kernel similarity function. Let $T = \{(x_1, y_1), \ldots, (x_m, y_m)\} \subset \mathbb{R}^d \times \{\pm 1\}$ be a set of labeled points, and $\alpha \in \mathbb{R}^m$ be a vector of m real numbers. Then for all $x \in \mathbb{R}^d$, we have that*

$$\sum_{i=1}^{m} \alpha_i y_i K(x_i, x) = \Big\langle \sum_{i=1}^{m} \alpha_i y_i \phi(x_i), \phi(x) \Big\rangle.$$

Because of this, if we let $w = \sum_{i=1}^m \alpha_i y_i \phi(x_i)$, then the kernel classifier $f^k_{T,\alpha}$ satisfies $f^k_{T,\alpha}(x) = f_{w,0}(\phi(x))$, where the latter classifier is the linear classifier in H with weight vector w.

The main idea behind Algorithm 3, is that it corresponds to running Algorithm 2 inside the embedded space of the kernel K. In particular, the kernel-perceptron update step precisely

corresponds to the dual-form of the perceptron-update step inside embedded space. It follows from Theorem C.4.2 that the following algorithm is identical to Algorithm 3.

Algorithm 10: Adversarial-Kernel-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathscr{D}^n$, Similarity function, K,
$w \leftarrow 0$
for $i = 1 \ldots n$ **do**
  $z = \arg\min_{||z-x||_p \leq r} y_i \langle w, \phi(z) \rangle$
  if $\langle y_i w, \phi(z) \rangle \leq 0$ **then**
    $w = w + y_i \phi(z)$
  end if
end for
return $f_{w,0} \circ \phi$

In particular, by comparing Algorithms 3 and 10, we have by Theorem C.4.2 that for all time steps t,

$$w = \sum_{(z,y) \in T} y \phi(z).$$

Therefore, to analyze the performance of Algorithm 3, it suffices to analyze Algorithm 10. However, we already have built to the tools for doing this: all of the results from Section C.3.1 apply to Algorithm 10 since the only difference is replacing $\mathbb{R}^d$ with H, the embedding space of K.

We now proceed by giving the corresponding assumptions on $\mathscr{D}$ needed for Theorem 3.4.5. We begin by first defining (K, r)-separability and K-robust margin, $\gamma_{r,K}$, the Kernel analogs of linear r-separability (Definition 3.2.12) and the robust margin (Definition 3.4.1).

Definition C.4.3. *For any $r > 0$, a distribution $\mathscr{D}$ over $\mathbb{R}^d \times \{\pm 1\}$ is (K, r)-**separable** if there exists a kernel classifier $f^K_{S,\alpha}$ such that $\mathscr{L}_r(f^K_{S,\alpha}, \mathscr{D}) = 0$.*

To define the K-robust margin, we will once again need the sets S_r^+ and S_r^- defined in equation 3.1 (top right of page 7). Recall that these sets denote the positively and negatively

labeled elements from $supp(\mathscr{D})$ *including* all adversarial perturbations of those points.

Definition C.4.4. *Let $\mathscr{D}$ be a (K,r)-separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then $\mathscr{D}$ has K-robust margin γ_r if γ_r is the largest real number such that there exists a kernel classifier $f^K_{T,\alpha}$, such that the following conditions hold.*

1. $\mathscr{L}_r(f^K_{T,\alpha}, \mathscr{D}) = 0$.
2. *Let ϕ, H be the embedding function/space of K, let $w = \sum_{(z,y)\in T} y\phi(z)$, and let $H_w = \{z \in H, \langle z, w\rangle = 0\}$ be the decision boundary in H of $f^K_{T,\alpha}$. Then for all $x \in S^+_r \cup S^-_r$, $\phi(x)$ has ℓ_2 distance at least γ^K_r from H_w inside H. That is,*

$$\inf_{x\in S^+_r \cup S^-_r} \inf_{z\in H_w} \sqrt{\langle \phi(x) - z, \phi(x) - z\rangle} = \gamma^K_r.$$

We now state the main theorem giving the performance of Algorithm 3.

Theorem C.4.5. *Let $\mathscr{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ such that the following conditions hold.*

1. *There exists $R > 0$ such that for all $x \in S^+_r \cup S^-_r$, $\langle \phi(x), \phi(x)\rangle \leq R^2$.*
2. *$\mathscr{D}$ is K, r-separable, and has K-robust margin $\gamma^K_r > 0$.*

Then for any $S \sim D^n$, if $f^k_{T,\alpha}$ denotes the classifier learned by Algorithm 3, then

$$\mathbb{E}_{S\sim\mathscr{D}^n}[\mathscr{L}_r(f^k_{T,\alpha}, \mathscr{D})] = O\left(\frac{(\gamma^K_r)^2}{R^2(n+1)}\right).$$

Proof. The key idea is to observe that Lemmas C.3.1 and C.3.2 both directly translate from Algorithm 9 to Algorithm 10. In particular, neither proof used the dimension, d, of $\mathbb{R}^d$, and consequently would equally apply to even an infinite dimensional Hilbet Space, H. Thus, the proof is completely analogous to the proof of Theorem C.3.2. □

Appendix D

Appendix for Chapter 4

D.1 Details for the proof of Theorem 4.4.2

Proof. We want to show that for any $m \in \mathbb{N}$, any learner on m samples must fail with constant probability. Toward this end, set $M = \binom{3m}{m}$, and let $Z_1^{(M)}, Z_2^{(M)}, \ldots, Z_M^{(M)}$ be subsets of $\mathbb{R}^d$ as described by Lemma 4.4.3 (we will drop the superscript in what follows). Let $\mathscr{M}$ denote the set of all subsets of $\{1, \ldots, 3m\}$ with exactly m elements. Associate with each Z_i a unique element of $\mathscr{M}$, thus allowing us to rename our subsets as $\{Z_T : T \in \mathscr{M}\}$. We will now construct a set of robustness regions U from these subsets. For $1 \leq i \leq 3m$, define

$$U_{x_i} = \cup_{T : i \in T} Z_T,$$

where x_i is an arbitrary point inside U_{x_i}. Note this is well-defined since the $Z^{(M)}$ are mutually disjoint.

By Lemma 4.4.3, it follows that if all x_i are given a label of -1, then any classifier $h \in \mathscr{H}_W$ satisfies that $h(z) = 1$ for some subset T and some $z \in Z_T$. However, this will imply that h lacks robustness on all $x \in \{x_i : i \in T\}$, meaning that there are at least m points among $\{x_1, \ldots, x_{3m}\}$ where h has robust loss 1. Furthermore, the second part of Lemma 4.4.3 implies that for any $T \in \mathscr{M}$, there exists a classifier h_T for which h_S is -1 over all $Z_{T'}$ for $T' \neq T$. This implies that h_T is robust at all x_i *except* for x_i with $i \in S$.

With these observations, we are now prepared to show that for any learner L, there exists a distribution D for which L has large expected robust loss. To do this, we use a standard lower bound technique found in [62] that was adapted to the robust setting in [9].

The idea will be to pick D to be the uniform distribution over a random subset of $2m$ points in $\{x_1, \ldots, x_{3m}\}$. We will then argue that because L only has access to m points from D, it won't be able to distinguish which subset D corresponds to, and this will lead to a large expected loss.

To this end, for any $T \in \mathscr{M}$, let D_T be the data distribution over $\mathbb{R}^d \times \{\pm 1\}$ where x is chosen at uniform from $\{x_i : i \notin T\}$ and y is always -1. We may assume without loss of generality that our learning algorithm, L, always outputs a classifier among the set $\{h_T : T \in \mathscr{M}\}$. This is because Lemma 4.4.3 implies that any classifier in $h \in \mathscr{H}_W$ has robust loss that is at least as bad some h_T (namely, if the decision boundary of h crosses Z_T).

Next, let $T, T' \in \mathscr{M}$ be arbitrary. By definition, h_T lacks robustness on all x_i with $i \in T$, and is perfectly accurate and robust at all other points. It follows that among the $2m$ points in the support of $D_{T'}$, there are $m - |T \cap T'|$ where h_T lacks robustness, implying the the loss of classifier h_T with respect to distribution $D_{T'}$ is $\frac{1}{2} - \frac{|T \cap T'|}{2m}$. Note that this implies that h_T has 0 robust loss over D_T (thus meeting the first stipulation of Theorem 4.4.2).

Finally, we bound the expected loss of the learner L with respect to a uniformly random choice of D_T. Let $\mathscr{M}$ also denote the uniform distribution over itself, and let $\mathscr{U}$ denote the uniform distribution over $\{1, 2, 3, \ldots, 3m\}$. Taking expectations over $T \sim \mathscr{M}$ and $S \sim D_T^m$, and letting $h_{L(S)}$ denote the classifier learned by L, we have that

$$\begin{aligned}
\mathbb{E}_{T \sim \mathscr{M}} \mathbb{E}_{S \sim D_T^m} \left[\ell_U(h_{L(S)}, D_T) \right] &= \mathbb{E}_{S \sim \mathscr{U}^m} \mathbb{E}_{T \sim (\mathscr{M}|S)} \left[\ell_U(h_{L(S)}, D_T) \right] \\
&= \mathbb{E}_{S \sim \mathscr{U}^m} \mathbb{E}_{T \sim \{T' : T' \in \mathscr{M}, S \cap T' = \emptyset\}} \left[\frac{1}{2} - \frac{|T \cap L(S)|}{2m} \right].
\end{aligned}$$

To bound the inner expectation, observe that since $|S| = m$, T' has a conditional distribution that is an arbitrary (at uniform) subset of at least $2m$ indices. Since $L(S)$ is fixed, it follows that the

probability that any element in $L(S)$ is an element of T' is at most $\frac{1}{2}$, meaning that the expected value of $|T \cap L(S)|$ is at most $\frac{|L(S)|}{2} = \frac{m}{2}$. Substituting this, we have that

$$\mathbb{E}_{T\sim\mathscr{M}}\mathbb{E}_{S\sim D_T^m}\left[\ell_U(h_{L(S)}, D_T)\right] \geq \mathbb{E}_{S\sim\mathscr{U}^m}\mathbb{E}_{T\sim\{T':T'\in\mathscr{M}, S\cap T'=\emptyset\}}\left[\frac{1}{2} - \frac{m}{4m}\right] = \frac{1}{4}.$$

By Markov's inequality, any random variable between 0 and 1 with expectation $\frac{1}{4}$ is strictly larger than $\frac{1}{8}$ with probability at least $\frac{1}{7}$. Since the loss above is bounded between 0 and 1, it follows that $\Pr_{T\sim\mathscr{M}}\Pr_{S\sim D_T}[\ell_U(h_{L(S)}, D) > \frac{1}{8}] \geq \frac{1}{7}$. Thus, for some $D = D_T$, the desired claim holds, finish the proof. □

D.2 Sample Oracle Lower Bounds

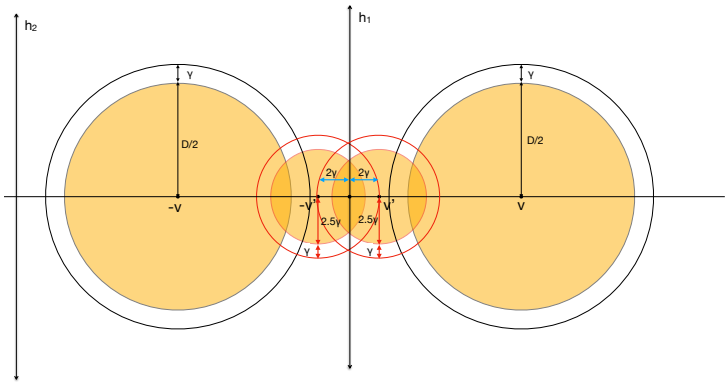


Figure D.1. Illustration for the sampling oracle lower bound in Proposition 4.5.3 in $\mathbb{R}^2$.

We now show a lower bound on the number of oracle calls required for tolerant learning in Ashtiani et al. [60]'s sample oracle model. We first recall the model itself for completeness, focusing on the case of $(\mathbb{R}^d, \ell_2)$ endowed with the standard Lebesgue measure for simplicity.

Definition D.2.1 (Sampling Oracle [60])**.** *Let $U : \mathbb{R}^d \to P(\mathbb{R}^d)$ be any perturbation function such that $U(x)$ has finite Lebesgue measure for all $x \in \mathbb{R}^d$. The sampling oracle $\mathcal{O}_U$ inputs any $x \in \mathbb{R}^d$, and outputs a sample y from the induced distribution on $U(x)$ under the Lebesgue measure.*

We prove that tolerant learning requires exponentially many calls to the sampling oracle.

Proposition 4.5.3. *For any $D > 10\gamma > 0$, there exists a hypothesis class $\mathscr{H}$ and a set of robustness regions, U such that the following holds. There exist constants ε and δ such that for any $n > 0$, any learner L on n samples that achieves*

$$\ell_U(L(S), \mathscr{D}) \leq \min_{h \in \mathscr{H}} \ell_{U^\gamma}(h, \mathscr{D}) + \varepsilon$$

with probability at least $1 - \delta$ must make at least $\Omega\left(\left(\frac{D}{\gamma}\right)^d\right)$ calls to the sampling oracle for some valid data distribution $\mathscr{D}$,

Proof of Proposition 4.5.3. Appealing to Yao's Minimax Principle, it is enough to find a class $\mathscr{H}$ and strategy for the adversary (over valid choices of perturbation sets and data distributions) such that any deterministic learner using at most $O((\frac{D}{\gamma})^d)$ oracle calls incurs at least constant error (ε) over the optimum in $\mathscr{H}$ with constant probability (δ).

With this in mind, fix $D_0 = D - 9\gamma$, let $r = 4\gamma$, and let e_1 denote the first canonical basis vector in $\mathbb{R}^d$. Our (marginal) data distribution will consist of two points in $\mathbb{R}^d$ $\left\{(\frac{D_0}{2} + 4\gamma)e_1, -(\frac{D_0}{2} + 4\gamma)e_1\right\}$. For the ease of notation, we denote $v \coloneqq (\frac{D_0}{2} + 4\gamma)e_1$. Note, $||v - (-v)||_2 = D_0 + 2r$. We now define the underlying hypothesis class $\mathscr{H}$ which consists of two linear classifiers $\mathscr{H} \coloneqq \{h_1, h_2\}$ such that $h_1 = sgn(\langle e_1, \cdot \rangle)$ and $h_2 = (\langle e_1, \cdot \rangle - D_0 - 4\gamma)$. Note that h_1 is a perpendicular bisector of the line segment joining v and $-v$, and h_2 is parallel to h_1 but biased to the left of v.

Finally, we construct two perturbation sets with bounded diameter U and V. Fix $v' = 2\gamma e_1$. We define balls of radius $r > 0$ for any given $x \in \mathbb{R}^d$ as $B_2(x, r) \coloneqq \left\{x' \in \mathbb{R}^d : ||x' - x||_2 \leq r\right\}$. First, we define a perturbation U and its γ-perturbed region U^γ as follows:

$$U \coloneqq \{U_v, U_{-v}\} \text{ where for any } x \in \{v, -v\}, U_x = B_2\left(x, \frac{D_0}{2}\right),$$
$$U^\gamma \coloneqq \left\{U_v^\gamma, U_{-v}^\gamma\right\} \text{ where for any } x \in \{v, -v\}, U_x^\gamma = B_2\left(x, \frac{D_0}{2} + \gamma\right)$$

Similarly, we define another perturbation set V and its γ-perturbed region V^γ:

$$V \coloneqq \{V_v, V_{-v}\} \text{ where for any } x \in \{v,-v\}, V_x = U_x \cup B_2\left(x', \frac{5\gamma}{2}\right),$$
$$V^\gamma \coloneqq \left\{V_v^\gamma, V_{-v}^\gamma\right\} \text{ where for any } x \in \{v,-v\}, V_x^\gamma = U_x^\gamma \cup B_2\left(x', \frac{7\gamma}{2}\right)$$

where $x' = v'$ or $-v'$ if $x = v$ or $-v$ respectively. We assume that the perturbation set for $\mathbb{R}^d \setminus \{v,-v\}$ is null for simplicity. Observe that $\bigcap_{x' \in \{v,-v\}} U_{x'} = \emptyset$ and so is the intersection of perturbations in U^γ. But, we note that $\bigcap_{x' \in \{v,-v\}} V_{x'} \neq \emptyset$. This entire construction is illustrated in Figure D.1.

We are now ready to describe the adversary's strategy, who chooses one of U or V independently with probability $1/2$, and employs a single fixed choice of data distribution $\mathscr{D}$ where $\Pr[Y = -1|-v] = \Pr[Y = 1|v] = 1$, and the marginal distribution is uniform over v and $-v$. Note that if the perturbation set is U, then h_1 is optimal as $\ell_U(h_1, \mathscr{D}) = 0$ whereas $\ell_U(h_2, \mathscr{D}) = 1/2$. On the other hand if V is chosen then h_2 is optimal as $\ell_V(h_2, \mathscr{D}) = \frac{1}{2}$ and $\ell_V(h_1, \mathscr{D}) = 1$. The idea is to show that the learner cannot distinguish between U and V with high probability, and thus cannot choose the right hypothesis. We note that since the data distribution is fixed and known to the learner, we only need to consider randomness over the sample oracle—labeled samples have no effect on the bound.

More formally, we split our analysis into two cases based on whether or not the learner draws an (oracle) sample in $V^\gamma \setminus U^\gamma$. First, note that conditioned on the fact that the learner draws no such sample, by construction the posterior probability of U is strictly higher than that of V. This means the learner's expected excess error is minimized by always outputting h_1 on such samples. On the other hand, if the learner observes a sample in $V^\gamma \setminus U^\gamma$, they can always achieve optimal error by outputting h_2.

Since the above learning rule minimizes the learner's expected excess error, it is enough

to bound the expected error of this rule:

$$\begin{aligned}\mathbb{E}_{Z,S\sim\mathscr{O}_Z}[OPT_Z - \ell_Z(\mathscr{A}(S),\mathscr{D})] &\geq \frac{1}{2}\Pr[S\subset U^\gamma \wedge Z = V] \\ &= \frac{1}{2}\Pr[Z=V]\Pr[S\subset U^\gamma | Z=V] \\ &= \frac{1}{4}\Pr[S\subset U^\gamma | Z=V]\end{aligned}$$

The key observation is then simply to notice that $\Pr[S\subset U^\gamma|Z=V]$ is constant whenever the learner draws at most $O((\frac{D}{\gamma})^d)$ oracle samples. This follows from the fact that under the induced distribution P_{V^γ} on V:

$$P_{V^\gamma}(V^\gamma \setminus U^\gamma) = \frac{\mu(V^\gamma\setminus U^\gamma)}{\mu(V^\gamma)} \leq \frac{\mu(B_2(v',\frac{7\gamma}{2}))}{\mu(U_v^\gamma)+\mu(B_2(v',\frac{7\gamma}{2}))} \leq \frac{(\frac{7}{2}\gamma)^d}{D_0^d}$$

where μ is the standard Lebesgue measure. Similarly we then have

$$P_{V^\gamma}(U^\gamma) \geq 1 - \frac{(\frac{7}{2}\gamma)^d}{D_0^d}$$

and finally that

$$\Pr[S\subset U^\gamma|Z=V] \geq \left(1-\frac{(\frac{7}{2}\gamma)^d}{D_0^d}\right)^{|S|}$$

which is at least some constant when $|S| \leq c(\frac{D_0}{\gamma})^d$ for some sufficiently small absolute constant $c<0$. Since $D_0 = D - 9\gamma$, there exists c' such that this holds when $|S|\leq c'(\frac{D}{\gamma})^d$ which implies the proposition.

□

We note that in Ashtiani et al. [60], the sampling oracle is defined more generally for any *doubling-measure* μ, that is any measure for which there exists some "doubling-constant" $C>0$

such that for all $x \in \mathbb{R}^d$ and $r \in \mathbb{R}^+$:

$$0 < \mu(B(x,2r)) \leq C\mu(B(x,r)) < \infty.$$

In this more general setting, one can prove a lower bound that scales with the doubling-constant (typically exponential in the associated doubling-dimension of the metric space) simply by appropriately increasing the concentration of measure on U^γ.

D.3 Robust VC for k points

In this section, we prove that the size-k perturbation sets only cost a $\log(k)$ factor over the VC dimension of the original class. To formalize this, we first recall a few basic definitions standard to the (adversarially robust) learning literature.

Definition D.3.1 (Robust Loss Class)**.** *Given a hypothesis $h : \mathscr{X} \to \{0,1\}$ and perturbation function $U : \mathscr{X} \to P(\mathscr{X})$, let $h^\ell_U : X \times \{0,1\}$ be the function over labeled samples measuring the robust loss of h:*

$$h^\ell_U(x,y) = \begin{cases} 0 & \text{if } \forall x' \in U(x) : h(x') = y \\ 1 & \text{else.} \end{cases}$$

The robust loss class of $(\mathscr{X}, \mathscr{H})$ is the hypothesis class over $\mathscr{X} \times \{0,1\}$ given by:

$$\mathscr{L}^U_{\mathscr{H}} := \{h^\ell_U : h \in \mathscr{H}\}.$$

We are interested in analyzing a standard complexity measure of the robust loss class called VC dimension

Definition D.3.2 (VC Dimension)**.** *The VC dimension of a hypothesis class $(\mathscr{X}, \mathscr{H})$ is the size of largest subset $S \subseteq \mathscr{X}$ such that $\mathscr{H}$ obtains all $2^{|S|}$ labelings on S. We say such a set is shattered by $\mathscr{H}$.*

We show the VC dimension of the robust loss class incurs at most $\log(k)$ blow-up over the original class.

Proposition D.3.3 (Overhead of Robust VC)**.** *Let $(\mathscr{X}, \mathscr{H})$ be a hypothesis class of VC-dimension d and $U : \mathscr{X} \to P(\mathscr{X})$ any perturbation function with support bounded by some $k \in \mathbb{N}$. Then the VC dimension of $\mathscr{L}^U_{\mathscr{H}}$ is at most $O(d \log(dk))$.*

This result was also independently communicated to us by Omar Montasser. The proof of Proposition D.3.3 relies on the classical Sauer-Shelah-Perles lemma, which we recall here for completeness.

Lemma D.3.4 (Sauer-Shelah-Perles [82, 83])**.** *Let $(\mathscr{X}, \mathscr{H})$ be a hypothesis class of VC-dimension d. Then for any finite subset $S \subseteq \mathscr{X}$, $\mathscr{H}$ obtains at most $O(|S|^d)$ distinct labelings on S.*

Proposition D.3.3 simply follows from using Sauer-Shelah-Perles to bound the total number of permissible patterns across perturbation sets of a sample in the loss space.

Proof of Proposition D.3.3. Let $m \in \mathbb{N}$ and assume there exists a sample $S = (x_1, y_1), \ldots, (x_m, y_m)$ that is shattered by $\mathscr{L}^U_{\mathscr{H}}$. We will show $m \leq O(d \log(kd))$. With this in mind, let $T = \bigcup_{i=1}^m U(x_i)$ denote the set of at most km points corresponding to the robustness regions of our sample. The key observation is the following (essentially trivial) claim:

Claim 1. *Any two $g^\ell_U, h^\ell_U \in \mathscr{L}^U_{\mathscr{H}}$ that give distinct labelings of S correspond to $g, h \in \mathscr{H}$ with distinct labelings of T.*

By robust shattering, there exist 2^m distinct labelings of S by $\mathscr{L}^U_{\mathscr{H}}$, so the above claim implies $\mathscr{H}$ must have 2^m distinct labelings of T. However the latter has at most $O((km)^d)$ labelings by VC dimension, so

$$2^m \leq O((km)^d) \Rightarrow m \leq O(d \log(dk))$$

by standard manipulations. Finally, we note the claim is immediate from definition, since the behavior of a function $h_U^\ell \in \mathscr{L}_{\mathscr{H}}^U$ on S depends only on the labels of its corresponding hypothesis $h \in \mathscr{H}$ on T by definition. □

D.4 Proof of Theorem 4.6.6

We begin by defining v_{ball}, which is the adversarial VC dimension when the robustness regions are all balls of a fixed radius. We start by precisely defining these robustness regions.

Definition D.4.1. *Let U^r be the set of robustness regions defined by $\{U_x^r = B(x,r)\}$, where $B(x,r)$ denotes the closed ball of ℓ_2-radius r centered at x.*

We now define the adversarial VC dimension of a set of classifiers $\mathscr{H}$ for a fixed set of regions, U^r.

Definition D.4.2. *Let $\mathscr{H}$ be a set of classifier. Then the adversarial VC dimension of $\mathscr{H}$ with respect to U^r is the maximum integer v, for which there exist v labeled points, $(x_1,y_1),\ldots,(x_r,y_r)$ so that for any subset $S \subset \{(x_1,y_1),\ldots,(x_r,y_r)\}$, there exists $h_S \in \mathscr{H}$ with*

$$\ell(h_S,(x_i,y_i) = \begin{cases} 0 & i \in S \\ & \\ 1 & i \notin S \end{cases}.$$

We denote this by v_{ball}^r.

Finally, we define v_{ball} as the maximum value of v_{ball}^r over all $r > 0$. Note that this quantity has been well studied – for example [49] shows that for linear classifiers, $v_{ball} = O(d)$.

Proving Theorem 4.6.6

We now turn our attention towards the proof. The key observation is that the main steps from the proof of Theorem 4.1.2 *perfectly* carry over. In particular, Lemma 4.6.7 exactly holds in this setting, and the argument given in the proof of Theorem 4.1.2 also holds provided that an

appropriate choice of V exists. The only issue arises from Lemma 4.6.8, which requires that $\mathscr{H}$ be regular. To remedy this, we now state and prove a different version of this lemma that uses a union of balls (of fixed radius) for V_x rather than a finite set of points.

Lemma D.4.3. *Let $\mathscr{H}$ by an arbitrary hypothesis class. For all $r \in [\frac{\varepsilon\delta\gamma}{7}, \gamma]$, let α, U^r and $U^{r-\alpha}$ be as described in the proof of Theorem 4.1.2. Then there exists a set of robustness regions $V^r = \{V_x^r : x \in \mathbb{R}^d\}$ satisfying the following two properties.*

1. *V_x^r is a union of $O\left(\left(\frac{D}{\varepsilon\delta\gamma}\right)^d\right)$ balls of radius , where D denotes the maximum diameter of U_x.*

2. *Let $\alpha = \frac{\varepsilon\delta\gamma}{7}$. For all labeled points (x,y) and for all classifiers $h \in \mathscr{H}$,*

$$\ell_{U^{r-\alpha}}(h,(x,y)) \leq \ell_{V^r}(h,(x,y)) \leq \ell_{U^r}(h,(x,y)).$$

Proof. Since $U_x^{r-\alpha}$ has diameter at most D, it follows that it can be covered with $O\left(\left(\frac{D}{\varepsilon\delta\gamma}\right)^d\right)$ balls of radius $\frac{\alpha}{2}$. We let V_x^r be any such cover that is minimal (meaning that (1.) is satisfied), meaning that each ball intersects $U_x^{r-\alpha}$. It follows that for all x, $U_x^{r-\alpha} \subseteq V_x^r \subseteq U_x^r$, which immediately implies (2.) and completes the proof. □

Finally, to prove Theorem 4.6.6, we note that the proof of Theorem 4.1.2 essentially works. The only differences are that instead of bounding the robust VC dimension of $\mathscr{H}$ with respect to V_X in terms of v, we must use v_{ball} as we are now considering unions of balls rather than points. As a detail, note that we are using the following minor modification of Proposition D.3.3 to bound the robust VC dimension of unions of balls using the robust VC dimension for balls.

Proposition D.4.4. *Let $(\mathscr{X}, \mathscr{H})$ be a hypothesis class whose robust loss class with respect to r-balls has VC dimension v_{ball}^r. Then the loss class of $(\mathscr{X}, \mathscr{H})$ with respect to perturbations that are a union of at most k r-balls has VC dimension at most $O(v_{ball}^r \log(v_{ball}^r k))$.*

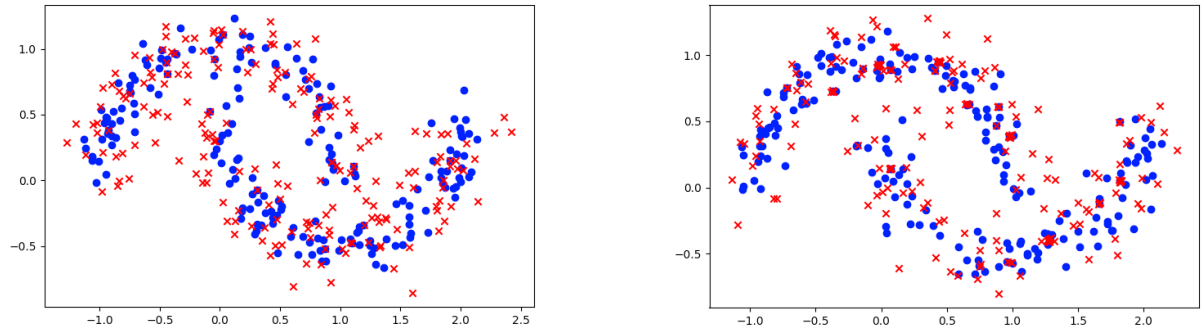
Proof. The proof is largely the same as D.3.3. Denote the original perturbation family as U, and the k-union perturbation family by U^k. Given a sample $S = (x_1, y_1), \ldots, (x_m, y_m)$, let C_i denote the centers of the at most k balls appearing in the perturbation set of x_i. It is enough to observe that any two distinct labelings of $S = (x_1, y_1), \ldots, (x_m, y_m)$ by $\mathscr{L}_{\mathscr{H}}^{U^k}$ correspond to distinct labelings of the extended sample $T = \bigcup_{i=1}^{m} (C_i, y_i)$ with respect to $\mathscr{L}_{\mathscr{H}}^{U}$, where (C_i, y_i) denotes the sample $\bigcup_{c \in C_i} (c, y_1)$. The bound then follows from the same double counting argument as in Proposition D.3.3. □

Appendix E

Appendix for Chapter 5

E.1 An Example over the Halfmoons dataset

In this section, we give an overview of our experiments over the Halfmoons dataset. Further details can be found in sec



(a) $\rho = 0.1$ **(b)** $\rho = 0.4$

Figure E.1. In the two panels above, the blue points are a training sample from p, and the red points are generated examples from q. The parameter ρ is the proportion of examples of q that are generated by q_{copy}, with the rest of the examples being drawn from $q_{underfit}$. As ρ increases, the rate of data-copying increases, which can be seen as the red points become increasingly clustered on top of a scattering of blue ones. However, due to $q_{underfit}$, there are still many red points that are relatively scattered from the blue points. At a global level, these effects average out making data-copying detection difficult for [3]'s method.

Our theoretical results show that given enough data, Algorithm 6 is guaranteed to detect data-copying. By contrast, the non-parametric test provided in [3] can only guarantee detection in cases in which data-copying *globally* occurs. For more local instances of data-copying, they

rely on k-means clustering to partition the input space into localized regions, and then run their global test over each region separately.

Their approach clearly cannot detect all forms of data-copying – a pathological generative distribution might copy in complex regions that are impossible to find using k-means clustering. However, for many practical examples considered in their paper, [3] demonstrated considerable success with this approach.

This motivates the following question:

> Do there exist natural data distributions over which Algorithm 6 offers a meaningful advantage?

We provide a partial answer to this question by experimentally comparing our approach with [3]'s over a simple example on the half moons dataset.

E.1.1 Experimental Setup

Data Distribution:

Our data distribution, p, is the Halfmoon dataset with Gaussian noise ($\sigma = 0.1$).

Generated Distribution:

Our generated distribution, q, is trained from an i.i.d sample of 2000 points from p, $S \sim p^{2000}$. Because our focus is on distinguishing *data-copy detection* algorithms, we design q to have a large amount of data-copying that is nevertheless subtle to detect. The key idea is to let q be a mixture of two distributions, q_{copy} and $q_{underfit}$. q_{copy} will be an egregious data copier, and $q_{underfit}$ will be designed to average away the effects of q_{copy}.

To construct q_{copy}, we first select a subset, $S' \subset S$, of 20 training examples. Then, we define q_{copy} to randomly output points from S' combined with a small amount of spherical noise (with radius 0.02). Thus, q_{copy} can be sampled from by sampling a point, x, from S' at uniform, and returning $x + \eta$ where η is drawn at uniform from a disk of radius 0.02.

To construct $q_{underfit}$, we combine our original data distribution, p, with a moderate amount of spherical noise (with radius 0.25). Thus, $q_{underfit}$ can be sampled from by first

sampling $x \sim p$, and returning $x + \eta$ where η is drawn at uniform from a disk of radius 0.25. This distribution is meant to represent a fairly noisy and thus underfit version of p.

Finally, we define q as a mixture of q_{copy} and $q_{underfit}$, with q outputting a point from q_{copy} with probability ρ. In total, we have

$$q = \rho \cdot q_{copy} + (1 - \rho) \cdot q_{good}.$$

We let, ρ, the weight of q_{copy} within the mixture, be a varying parameter that gives rise to different generated distributions. Intuitively, the larger ρ is, the higher the data-copying rate. This is illustrated in Figure E.1. In the both panels, we plot a sample of 200 training points p along with 200 points from q. In the left panel, we let $\rho = 0.1$ in the right, we use $\rho = 0.4$. Although both cases show examples of data-copying, the right panel shows a visibly higher level of it. This is expected, as it is drawn from a distribution in which q_{copy} is much more likely to be queried.

Data-copying Detection Algorithms:

We run our algorithm, Data_Copy_Detect, on (S, q), We fix $\lambda = 20$ and $\gamma = 0.00025$ as constants for data-copy detection. λ represents a healthy level of data-copying, and $\gamma = 0.00025$ ensures that our condition for 'copying' is quite stringent. Full details of our implementation (including our practical choices for parameters such as b and m) are given in Appendix 5.5.2.

For comparison, we also include an implementation of [3]'s algorithm with varying amounts of clusters being used for the initial k-means clustering. To avoid confusion with the intrinsic dimension, k, we let c denote the number of clusters, and consider $c \in \{1, 5, 10, 20\}$.

E.1.2 Results

The results are summarized in Table E.1, with each column corresponding to a given choice of p, q (determined by the parameter ρ), and each row corresponding to a separate data-copying detection algorithm. As a baseline, we include the case where $q = p$ (meaning we have

a perfect generated distribution) in the first column.

We run our algorithm with parameters λ and γ fixed as 20 and 0.00025 in all cases. For [3]'s algorithm, we consider their data-copy detection score over the most egregious cluster.

Although our algorithm outputs real number estimates of the true data-copying rate, cr_q, [3]'s algorithm outputs a score indicating the statistical significance of their metric under a null hypothesis of no data-copying occurring. To facilitate a simple comparison between our methods, for all algorithms, we simply output a simple yes or no to indicate whether our results were statistically significant up to the $p = 0.05$ level. We include full results of our experiments along with several extensions (with varying parameters) in section E.1.3.

As expected, neither of our algorithms detect data-copying on the baseline, $q = p$. However, in all other cases, our algorithm successfully detects data-copying. On the other hand, for the smaller values of ρ, [3]'s does not. Their algorithm is only able to achieve detection when the weight of $\rho = 0.4$, and even in this case they are unable to consistently do so.

These results match the simple intuition of our algorithms. As seen in Figure E.1, the red data is sometimes very close to the blue data (when it comes from q_{copy}) but at other times fairly distant (when it comes from $q_{underfit}$). These effects have a strong canceling effect in [3]'s test. However, our test is able to adjust for this by considering each training example separately.

Table E.1. Statistical Significance of data-copying Rates over Halfmoons

Algo	**q = p**	ρ **= 0.1**	**0.2**	**0.3**	**0.4**
Ours	no	yes	yes	yes	yes
c = 1	no	no	no	no	no
c = 5	no	no	no	no	yes
c = 10	no	no	no	no	yes
c = 20	no	no	no	yes	yes

E.1.3 Further Experimental Details

We begin by reviewing the definitions of p and q. p is the Halfmoons dataset with Gaussian noise ($\sigma = 0.1$). To define q, we have a mixture of two distributions, q_{copy} and $q_{underfit}$, which are defined as follows.

We draw $S \sim p^{2000}$ i.i.d, and then randomly select $S' \subset S$ with $|S'| = 20$. These points will form a basis for the support of q_{copy}. To sample $x \sim q_{copy}$, we take the following two steps.

1. Sample $z \sim S'$ at uniform.
2. Sample $\eta \sim U(B(0, 0.02))$, where $U(B(0,r))$ denotes the uniform distribution over the ball of radius r.
3. Output $x = z + \eta$.

q_{copy} can be thought of as an egregious data memorizer that injects a small amount of noise to give its inputs some (paltry) variety.

By contrast, to sample $x \sim q_{underfit}$, we do the following:

1. Sample $z \sim p$.
2. Sample $\eta \sim U(B(0, 0.25))$.
3. Output $x = z + \eta$.

In this case, the larger amount of noise serves to induce *underfitting*, in which q_{copy} does not assign the support of p enough probability mass.

Finally, to sample from q, we do the following.

1. With probability ρ, sample $x \sim q_{copy}$.
2. With probability $1 - \rho$, sample $x \sim q_{underfit}$.

[3]'s test:

Their test works as follows. Let S denote the original training sample, Q denote a sample of generated examples, with $Q \sim q^n$, and P denote a fresh set of test examples, with $P \sim p^n$. They then check to see if Q is systematically closer to S than P, (thus suggesting data copying). To do so, they use a statistical test as follows:

1. Let $S = \{x_1, x_2, \ldots, x_n\}$, $P = \{y_1, y_2, \ldots, y_n\}$, $Q = \{z_1, z_2, \ldots, z_n\}$.

2. Let Δ denote the number of pairs (i, j) for which $d(y_i, S) < d(z_j, S)$. A large value of Δ indicates that a *small* amount of data copying, as it implies that Q is further from S than P. A small value of Δ indicates a **large** amount of data copying.

3. Reflecting this, let $Z = \frac{\Delta - \frac{n^2}{2}}{\sqrt{\frac{n^2(2n+1)}{12}}}$. This gives a Z-score of Δ. [3] show that, $p = q$, then the probability of results as significant as $Z < -5$ would be at most the probability of getting a -5σ event when sampling from a Gaussian. We use $Z < -3$ to indicate *statistically significant results*, and output the corresponding P-values ($P = 0.0027$ being significant) in our results.

Finally, to account for data copying occurring within specific regions, [3] perform a preprocessing step in which they cluster the training data, S into c regions using k-means clustering. They then run their test separately on each region by assigning points from P and Q into the regions containing them. We output the *lowest* Z-score over any region, and vary the number of clusters with $c = 1, 5, 10, 20$.

Our test:

We run Algorithm 6 with input (S, q) with a few adjustments.

1. We directly set $m = 200,000$. While the theoretical value of m is significantly higher (growing $O(n^2)$), we note that this is primarily done for achieving theoretical guarantees. In practice, often a much lower amount of data is needed.

2. For $Est(x,r,S)$, we set $b = 400$, which is a bit lower than the theoretically predicted value. As for m, we do this because for practical (and well-behaved) datasets, $Est(x,r,S)$ converges much more quickly than theory suggests.

3. We set $\lambda = 20$ and $\gamma = \frac{1}{4000}$, giving relatively stringent conditions on data copying.

Finally, our test outputs, $\hat{cr}_q$, which is an estimate of the data copy rate. Technically, any non-zero of $\hat{cr}_q$ indicates a degree of data copying. To facilitate a more direct comparison with [3], we convert our results into statistical tests by doing the following.

1. We compute $\hat{cr}_p$, which is an estimate for the data copying rate when the generated distribution exactly equals p over 1000 different instances (each instance corresponding to a freshly drawn training set S).

2. We then compute $\hat{cr}_q$ when q is as above.

3. We finally output the fraction of the time that $\hat{cr}_p > \hat{cr}_q$, thus giving us a P-value by giving us the rate at which the null-hypothesis gives results as significant as those that we observe.

Results:

We give a more complete version of Table E.1, with the P-values themselves being outputted in the table. For consistency, we output the median P-value obtained over 10 runs for each experiment.

Table E.2. P-values of data-copying Rates over Halfmoons

Algo	$\mathbf{q = p}$	$\boldsymbol{\rho = 0.1}$	**0.2**	**0.3**	**0.4**
Ours	1.000	0.000	0.000	0.000	0.000
$\mathbf{c = 1}$	0.5412	1.000	1.000	0.858	0.026
$\mathbf{c = 5}$	0.113	0.976	0.780	0.081	0.007
$\mathbf{c = 10}$	0.090	0.814	0.294	0.013	0.000
$\mathbf{c = 20}$	0.035	0.279	0.093	0.005	0.000

We also remark that the computed data-copying rates by our algorithm exactly match the value of ρ in all cases (up to 3 decimal points).

E.2 Estimating k

The main idea of our method is to simply pick any point x_i in the training sample, $S = \{x_1, x_2, \ldots, x_n\}$, choose two small balls centered at x_i, and then measure the ratio of their probability masses as well as their radii. For sufficiently small balls, these ratios will be related by a power of k, and we can consequently just solve for an estimate of k, $\hat{k}$. Finally, since for our purposes it is extremely important that our estimate be *exactly* correct, we round $\hat{k}$ to the nearest integer. While this clearly fails in cases that k is not an integer, for most distributions k precisely equals the dimension of the underlying data manifold (see for example Proposition 5.4.2). These steps are enumerated in the following algorithm, $Estimate\_k(S)$.

Algorithm 11: $Estimate\_k(S)$

$n \leftarrow |S|$
Pick $x \in S$ arbitrarily.
$b \leftarrow \frac{64(d+2)\ln\frac{16n}{\delta}}{\varepsilon^2}$.
$r_* = \min\{r : |S \cap B(x,r)| = 2b\}$.
$s_* = \min\{s : |S \cap B(x,s)| = b\}$
$\hat{k} = round\left(\frac{1}{\log_2 \frac{r_*}{s_*}}\right)$
Return $\hat{k}$.

We now give sufficient conditions under which Algorithm 11 successfully recovers k.

Proposition E.2.1. *Let p be an k-regular distribution, and let $\delta > 0$ be arbitrary. Let $\phi = \frac{1}{2k}$. Then there exists a constant C such that if*

$$n \geq C \frac{d \ln \frac{d}{\delta \phi p_\phi}}{\phi^2 p_\phi},$$

with probability at least $1 - \delta$ over $S \sim p^n$, $Estimate\_k(S) = k$.

Proof. We begin by first applying standard uniform convergence over ℓ_2 balls in $\mathbb{R}^d$ (which have a VC dimension of at most $d+2$). To this end, let

$$\beta_n = \sqrt{\frac{4(d+2)\ln\frac{16n}{\delta}}{n}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\delta$ over $S \sim p^n$, for all $x \in \mathbb{R}^d$ and all $r > 0$,

$$\frac{|S \cap B(x,r)|}{n} - \beta_n\sqrt{\frac{|S \cap B(x,r)|}{n}} \leq p(B(x,r)) \leq \frac{|S \cap B(x,r)|}{n} + \beta_n^2 + \beta_n\sqrt{\frac{|S \cap B(x,r)|}{n}}. \quad \text{(E.1)}$$

Next, assume that

$$n \geq \frac{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)}{\phi^2 p_\phi}. \quad \text{(E.2)}$$

It is clear that for an appropriate constant, we have $n = O\left(\frac{d\ln\frac{d}{\delta\phi p_\phi}}{\phi^2 p_\phi}\right)$. Thus, it suffices to show that if Equation E.1 holds, then $\hat{k} = k$ (as the former holds with probability $1-\delta$ over S). We now show the following claim.

Claim: Let $r > 0$ be any radius with $|S \cap B(x,r)| \geq b$. Then

$$\left(1+\frac{\phi}{9}\right)^{-1} \leq \frac{|S \cap B(x,r)|}{np(B(x,r))} \leq \left(1+\frac{\phi}{9}\right).$$

Proof. From the definition of b, we have that

$$\frac{b}{n} = \frac{400(d+2)\ln\frac{16n}{\delta}}{n\phi^2} = \frac{100\beta_n^2}{\phi^2}. \quad \text{(E.3)}$$

Let $c = \sqrt{\frac{b'}{n\beta_n^2}}$. Then $b' \geq b$ implies that $c \geq \frac{10}{\phi}$. It follows that

$$\frac{c+1}{c^2} \leq \frac{1}{c-1} \leq \frac{\phi}{9}. \tag{E.4}$$

Substituting Equations E.3 and E.4 into Equation E.1, we have

$$\begin{aligned} \frac{b'}{np(B(x,r))} &\geq \frac{\frac{b'}{n}}{\frac{b'}{n} + \beta_n^2 + \beta_n\sqrt{\frac{k'}{n}}} \\ &= \frac{c^2}{c^2+1+c} \\ &= \left(1 + \frac{c+1}{c^2}\right)^{-1} \\ &\geq \left(1 + \frac{\phi}{9}\right)^{-1} \end{aligned} \tag{E.5}$$

and

$$\begin{aligned} \frac{b'}{np(B(x,r))} &\leq \frac{\frac{b'}{n}}{\frac{b'}{n} - \beta_n\sqrt{\frac{b'}{n}}} \\ &= \frac{c^2}{c^2-c} \\ &= 1 + \frac{1}{c-1} \\ &\leq 1 + \frac{\phi}{9}, \end{aligned} \tag{E.6}$$

Together, Equations E.5 and E.6 imply our claim. □

We now return to the proof of Proposition 5.4.3. We now show that

$$p(B(x,s_*) \leq p(B(x,r_*)) \leq p_\phi.$$

To do so, we first bound β_n^2 as follows. We have,

$$\begin{aligned}
\beta_n^2 &= \frac{4(d+2)\ln(16n/\delta)}{n} \\
&= 4(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)\right)\frac{\phi^2 p_\phi}{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)} \\
&\leq 8(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)\frac{\phi^2 p_\phi}{1776(d+2)\ln\left(\frac{28416(d+2)}{\delta\phi^2 p_\phi}\right)} \\
&= \frac{p_\phi\phi^2}{222}.
\end{aligned} \tag{E.7}$$

Next, by Equations E.1 and E.7 along with the fact that $b = \frac{100\beta_n^2}{\phi^2}$ (Equation E.3) that

$$\begin{aligned}
p(B(x,r_*)) &\leq \frac{|S\cap B(x,r_*)|}{n} + \beta_n^2 + \beta_n\sqrt{\frac{|S\cap B(x,r_*)|}{n}} \\
&= \frac{2b}{n} + \beta_n^2 + \beta_n\sqrt{\frac{2b}{n}} \\
&= \beta_n^2\left(\frac{200}{\phi^2} + 1 + \frac{20}{\phi}\right) \\
&\leq \frac{p_\phi\phi^2}{222}\frac{221}{\phi^2} = p_\phi.
\end{aligned}$$

It follows from Definition 5.4.1 that

$$\left(1+\frac{\phi}{3}\right)^{-1}\frac{p(B(x,r_*))}{p(B(x,s_*))} \leq \frac{r_*^k}{s_*^k} \leq \left(1+\frac{\phi}{3}\right)\frac{p(B(x,r_*))}{p(B(x,s_*)}. \tag{E.8}$$

However, $|S\cap B(x,s_*)| = b$ and $|S\cap B(x,r_*)| = 2b$, which means that we can safely apply our claim to both of these cases. By substituting Equations E.5 and E.6 (for both r_*, s_*) into Equation E.8, along with the fact that $\left(1+\frac{\phi}{3}\right)\left(1+\frac{\phi}{9}\right) \leq \left(1+\frac{\phi}{2}\right)$, it follows that

$$\left(1+\frac{\phi}{2}\right)^{-1} \leq \frac{r_*^k}{2s_*^k} \leq \left(1+\frac{\phi}{2}\right) \tag{E.9}$$

Finally, by taking logs of Equation E.9 and simplifying, we have that

$$\frac{k}{1+\log_2\left(1+\frac{\phi}{2}\right)} \leq \frac{1}{\log_2 \frac{r_*}{s_*}} \leq \frac{k}{1-\log_2\left(1+\frac{\phi}{2}\right)}$$

It consequently suffices to show that k is the unique integer between $\frac{k}{1+\log_2(1+8\varepsilon)}$ and $\frac{k}{1-\log_2(1+2\varepsilon)}$. However, this is simply a result of the assumption that $\phi = \frac{1}{2k}$ and standard manipulations, which completes the proof. □

E.3 Proofs

All proofs to theorems and propositions in the main body are in this section. For each result, we include a restatement for convenience.

E.3.1 Proof of Theorem 5.2.3

We prove a stronger version of Theorem 5.2.3.

Theorem E.3.1 (Theorem 5.2.3)**.** *Let $1 < \lambda$ and $\gamma > 0$. Let σ_n be a sequence of bandwidths and K be any regular kernel function. For any $n > 0$ there exists a probability distribution π with full support over $\mathbb{R}^d$ such for any $S \sim \pi^n$, a KDE trained with bandwidth σ_n and kernel function K has data-copy rate $cr_q \geq \frac{1}{2}$.*

We begin by giving necessary conditions for a kernel K to be regular.

Definition E.3.2. *A kernel function, $K : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$ is regular if it satisfies the following conditions.*

1. *K is radially symmetric. That is, there exists $h : \mathbb{R} \to \mathbb{R}$ such that $K(x) = h(||x||)$.*
2. *K is regularized. That is, $\int_{\mathbb{R}^d} K(x)dx = 1$.*
3. *K decays to 0. That is, $\lim_{t\to\infty} h(t) = \lim_{t\to-\infty} h(t) = 0$.*

It is well known that under suitable choices of σ_n and several technical assumptions that a regular KDE converges towards the true data distribution in the large sample limit. We now prove Theorem 5.2.3.

Proof. Fix any n, and for convenience let denote σ_n by σ. Because K is non-negative, by condition 2. of Definition E.3.2, there exists $R > 0$ such that $\int_{||x|| \leq R} K(x)dx = \frac{1}{2}$. Let

$$D = R\sigma \left(\max \left(2n\lambda, \frac{1}{\gamma} \right) \omega_d \right)^{1/d},$$

where ω_d denotes the volume of the unit ball in d dimensions. We let π denote the uniform distribution over $[0,D]^d$, and claim that this suffices.

Let $S \sim \pi^n$ be a training sample, with $S = \{x_1, x_2, \ldots, x_n\}$, and let q be a KDE trained from S with bandwidth σ and kernel function K. Suppose $x \sim q$ satisfies that $x \in B(x_i, R\sigma)$. We claim that q (λ, γ)-copies x.

To see this, it suffices to bound $\pi((B(x_i, R\sigma))$ and $q(B(x_i, R\sigma))$. The former quantity satisfies

$$\begin{aligned}
\pi((B(x_i, R\sigma)) &\leq \frac{vol(B(x_i, R\sigma))}{D^d} \\
&= \frac{\omega_d (R\sigma)^d}{D^d} \\
&= \frac{1}{\max\left(2n\lambda, \frac{1}{\gamma}\right)} \\
&\leq \min\left(\gamma, \frac{1}{2n\lambda}\right),
\end{aligned}$$

which implies that the third condition of Definition 5.2.1 is met. Meanwhile, $q((B(x_i, R\sigma))$ can

be bounded as

$$\begin{aligned} q((B(x_i, R\sigma)) &= \int_{B(x_i,R\sigma)} \frac{1}{n\sigma} \sum_{j=1}^{n} K\left(\frac{x-x_j}{\sigma}\right) dx \\ &\geq \int_{B(x_i,R\sigma)} \frac{1}{n\sigma} K\left(\frac{x-x_i}{\sigma}\right) dx \\ &= \int_{||u||\leq R} \frac{1}{n} K(u) du \\ &\geq \frac{1}{2n}, \end{aligned}$$

which implies that $q((B(x_i, R\sigma)) \geq \lambda p(B(x_i, R\sigma))$ giving the second condition of Definition 5.2.1. Thus, it follows that q (λ, γ)-copies all $x \in B(x_i, R\sigma)$. It consequently suffices to bound $q\left(\bigcup_{i=1}^{n} B(x_i, R\sigma)\right)$.

To do so, let η denote the probability distribution over $\mathbb{R}^d$ with probability density function $\eta(x) = \frac{1}{\sigma} K(\frac{x}{\sigma})$, and let $\hat{q}$ denote the probability density function induced by the following random process:

1. Select $1 \leq i \leq n$ at uniform.
2. Select $x \sim \eta$.
3. Output $x + x_i$.

The key observation is that $\hat{q}$ has precisely the same density function as q – qs density function

is clearly a convolution of selecting x_i and then adding $x \sim \eta$. Applying this, we have

$$\begin{aligned}
\Pr_{x\sim q}\left[x \in \bigcup_{i=1}^{n} B(x_i, R\sigma)\right] &= \Pr_{x\sim \hat{q}}\left[x \in \bigcup_{j=1}^{n} B(x_j, R\sigma)\right] \\
&= \frac{1}{n}\sum_{i=1}^{n} \Pr_{x\sim\tau}\left[x \in \left(\bigcup_{j=1}^{n} B(x_j, R\sigma) - x_i\right)\right] \\
&\geq \frac{1}{n}\sum_{i=1}^{n} \Pr_{x\sim\tau}\left[x \in (B(x_i, R\sigma) - x_i)\right] \\
&= \int_{B(0,R\sigma)} \tau(x)dx \\
&= \int_{B(0,R\sigma)} \frac{1}{\sigma} K\left(\frac{x}{\sigma}\right) dx \\
&= \int_{B(0,R)} K(u)du \\
&\geq \frac{1}{2},
\end{aligned}$$

completing the proof.

□

E.3.2 Proof of Proposition 5.4.2

Proposition E.3.3 (Proposition 5.4.2)**.** *Let p be a probability distribution with support precisely equal to a smooth, compact, k-dimensional sub-manifold of $\mathbb{R}^d$, M. Additionally, suppose that p has a continuous density function over M. Then it follows that p is k-regular.*

To prove this, we begin with the following lemma.

Lemma E.3.4. *Let $k > 0$ be a constant. Let p be a probability distribution for which the following properties hold:*

1. The map $supp(p) \times \mathbb{R}^+ \to R^+$ defined by $(x, r) \mapsto p(B(x, r))$ is continuous.

2. The map $supp(p) \to \mathbb{R}^+$ defined by $x \mapsto \lim_{r\to 0} \frac{p(B(x,r)}{r^k}$ is well defined, continuous, and strictly positive over its domain.

3. p has compact support.

Then p is k-regular.

Proof. The map $r \to r^k$ is clearly continuous. It follows by properties (1.) and (2.), the following is a continuous map: $F : supp(p) \times \mathbb{R}^{\geq 0} \to \mathbb{R}^+$ where

$$F(x,r) = \begin{cases} \frac{p(B(x,r))}{r^k} & r > 0 \\ \lim_{s \to 0} \frac{p(B(x,s))}{s^k} & r = 0, \end{cases}.$$

Next, fix $\varepsilon > 0$, as arbitrary. We desire to show that p_ε exists for which the conditions of Definition 5.4.1 hold. Without loss of generality, we can assume $\varepsilon < 1$, as the case $\varepsilon \geq 1$ can easily be handled by just using p_ε for a smaller value of ε.

For any $x > 0$, since F is continuous, there exists $\rho_x > 0$ such that for any $x', \in B(x, \rho_x)$ and $r \leq \rho_x$,

$$|F(x',r) - F(x,0)| < F(x,0)\frac{\varepsilon}{9}.$$

It follows for any such x' that

$$p(B(x',\rho_x)) = F(x,\rho_x)\rho_x^k \geq (F(x,0))(1-\frac{\varepsilon}{9}), \tag{E.10}$$

and for any $0 < s < r < \rho_x$, we have

$$\begin{aligned} \frac{p(B(x',r))}{r^k} &= F(x',r) \\ &\leq F(x,0)(1+\frac{\varepsilon}{9}) \\ &\leq F(x',s)\frac{1+\frac{\varepsilon}{9}}{1-\frac{\varepsilon}{9}} \\ &\leq F(x',s)\left(1+\frac{\varepsilon}{3}\right), \end{aligned}$$

and

$$\begin{aligned}\frac{p(B(x',r))}{r^k} &= F(x',r)\\ &\geq F(x,0)(1-\frac{\varepsilon}{9})\\ &\geq F(x',s)\frac{1-\frac{\varepsilon}{9}}{1+\frac{\varepsilon}{9}}\\ &\geq F(x',s)\left(1+\frac{\varepsilon}{3}\right)^{-1},\end{aligned}$$

which together imply that

$$\left(1+\frac{\varepsilon}{3}\right)^{-1}\frac{p(B(x,s))}{s^k} \leq \frac{p(B(x,r))}{r^k} \leq \left(1+\frac{\varepsilon}{3}\right)\frac{p(B(x,s))}{s^k}. \tag{E.11}$$

Finally, observe that the balls $B(x,r_x)$ cover the support of p. Since $supp(p)$ is compact, it follows that there exists a finite sub-cover of such balls, C. We finally let $p_\varepsilon = \min_{B(x,r_x)\in C} F(x,0)(1-\frac{\varepsilon}{9})$. It then follows by Equations E.10 and E.11, that p has met the criteria necessary for p to be k-regular, as desired. □

We are now prepared to prove Proposition 5.4.2.

Proof. It suffices to show that the conditions of Lemma E.3.4 hold. Conditions 1. and 3. immediately hold since the probability mass of the surface (i.e. points on the boundary) of a ball $B(x,r)$ will be 0 as its intersection with M would be a $(k-1)$-dimensional manifold.

Thus, it remains to verify condition 2. For any $x,y \in M$, let $d_M(x,y)$ denote the geodesic distance between x and y (with $||x-y||$ still denoting their euclidean distance in $\mathbb{R}^d$ as M is embedded in $\mathbb{R}^d$). Since M is a smooth, compact manifold, it follows that for any $x \in M$,

$$\lim_{r\to 0}\sup_{||x-y||=r}\frac{||x-y||}{d_M(x,y)} = 1.$$

In other words, at a small scale, the geodesic distance and the Euclidean distance converge. It

follows that

$$\lim_{r\to 0}\frac{p(B(x,r))}{r^k}=\lim_{s\to 0}\frac{p(B_M(x,s))}{s^k},$$

where $B_M(x,s)$ denotes the geodesic ball of radius s centered at x on M. However, the latter quantity is precisely equal to the density function over M (up to a constant factor, since $\lim_{s\to 0}\frac{vol_M(B_M(x,s))}{s^k}=\omega_k$, where ω_k is the volume of the k-sphere). Since by assumption our density function is continuous and non-zero everywhere on the manifold, it follows that the map above must be well defined and continuous giving us condition 2. of Lemma E.3.4, as desired. □

E.3.3 Proof of Proposition 5.4.3

Proposition E.3.5 (Proposition 5.4.3)**.** *Let p be an k-regular distribution, and let $\varepsilon>0$ be arbitrary. Then if $n=O\left(\frac{d\ln\frac{1}{\delta\varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}\right)$ with probability at least $1-\delta$ over $S\sim p^n$, for all $x\in\mathbb{R}^d$ and $r>0$,*

$$\left(1+\frac{\varepsilon}{2}\right)^{-1}p(B(x,r))\le Est(x,r,S)\le\left(1+\frac{\varepsilon}{2}\right)p(B(x,r)). \tag{E.12}$$

Proof. We begin by first applying standard uniform convergence over ℓ_2 balls in $\mathbb{R}^d$ (which have a VC dimension of at most $d+2$). To this end, let

$$\beta_n=\sqrt{\frac{4(d+2)\ln\frac{16n}{\delta}}{n}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\delta$ over $S\sim p^n$, for all $x\in\mathbb{R}^d$ and all $r>0$,

$$\frac{|S\cap B(x,r)|}{n}-\beta_n\sqrt{\frac{|S\cap B(x,r)|}{n}}\le p(B(x,r))\le\frac{|S\cap B(x,r)|}{n}+\beta_n^2+\beta_n\sqrt{\frac{|S\cap B(x,r)|}{n}}. \tag{E.13}$$

Next, assume that

$$n \geq \frac{888(d+2)\ln\left(\frac{14208(d+2)}{\delta \min(\varepsilon,1)^2 p_\varepsilon}\right)}{\min(\varepsilon,1)^2 p_\varepsilon}. \tag{E.14}$$

It is clear that for an appropriate constant, we have $n = O\left(\frac{d\ln\frac{d}{\delta\varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}\right)$. Thus, it suffices to show that if Equation E.13 holds for all x, r, then the desired bound, Equation E.12, does as well.

To this end, let x, r be arbitrary, and let b be as defined in Algorithm 5. Let $b' = |S \cap B(x,r)|$ be the number of elements from S in $B(x,r)$. Then we have two cases.

Case 1: $b' \geq b$

It follows from Algorithm 5 that $Est(x,r,S) = \frac{b'}{n}$. We now set b as

$$\frac{b}{n} = \frac{400(d+2)\ln\frac{16n}{\delta}}{n\min(\varepsilon,1)^2} = \frac{100\beta_n^2}{\varepsilon^2}, \tag{E.15}$$

which clearly obeys the desired asymptotic bound given in Algorithm 5. Let $c = \sqrt{\frac{b'}{n\beta_n^2}}$. Then $b' \geq b$ implies that $c \geq \frac{10}{\min(\varepsilon,1)}$. It follows that

$$\frac{c+1}{c^2} \leq \frac{1}{c-1} \leq \frac{\min(\varepsilon,1)}{9}. \tag{E.16}$$

Substituting Equations E.15 and E.16 into Equation E.13, we have

$$\begin{aligned}
\frac{Est(x,r,S)}{p(B(x,r))} &\geq \frac{\frac{b'}{n}}{\frac{b'}{n} + \beta_n^2 + \beta_n\sqrt{\frac{b'}{n}}} \\
&= \frac{c^2}{c^2+1+c} \\
&= \left(1 + \frac{c+1}{c^2}\right)^{-1} \\
&\geq \left(1 + \frac{\min(\varepsilon,1)}{9}\right)^{-1}
\end{aligned} \tag{E.17}$$

and

$$\begin{aligned}\frac{Est(x,r,S)}{p(B(x,r))} &\leq \frac{\frac{b'}{n}}{\frac{b'}{n}-\beta_n\sqrt{\frac{b'}{n}}} \\ &= \frac{c^2}{c^2-c} \\ &= 1+\frac{1}{c-1} \\ &\leq 1+\frac{\min(\varepsilon,1)}{9}.\end{aligned} \tag{E.18}$$

Together, Equations E.17 and E.18 imply that $Est(x,r,S)$ is sufficiently accurate.

Case 2: $b' < b$

We begin by bounding β_n^2 in terms of p_ε. We have,

$$\begin{aligned}\beta_n^2 &= \frac{4(d+2)\ln(16n/\delta)}{n} \\ &= 4(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\varepsilon,1)^2 p_\varepsilon}\ln\left(\frac{14208(d+2)}{\delta\min(\varepsilon,1)^2 p_\varepsilon}\right)\right)\frac{\min(\varepsilon,1)^2 p_\varepsilon}{888(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\varepsilon,1)^2 p_\varepsilon}\right)} \\ &\leq 8(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\varepsilon,1)^2 p_\varepsilon}\right)\frac{\min(\varepsilon,1)^2 p_\varepsilon}{888(d+2)\ln\left(\frac{14208(d+2)}{\delta\min(\varepsilon,1)^2 p_\varepsilon}\right)} \\ &= \frac{p_\varepsilon\min(\varepsilon,1)^2}{111}.\end{aligned} \tag{E.19}$$

Now, let r_* be as defined in Algorithm 5. Then $|S\cap B(x,r_*)| = b$. Our main idea will be to show that $p(B(x,r_*) \leq p_\varepsilon$, and then use Equations E.17 and E.18 for r_* (which is possible since $|S\cap B(x,r_*)| = b$) along with the definition of p_ε (Definition 5.4.1) to bound $Est(x,r,S)$ in terms of $p(B(x,r))$. To this end, we have by Equations E.13 and E.19 along with the fact that

$b = \frac{100\beta_n^2}{\min(\varepsilon,1)^2}$ (Equation E.15) that

$$\begin{aligned} p(B(x,r_*)) &\leq \frac{|S \cap B(x,r_*)|}{n} + \beta_n^2 + \beta_n \sqrt{\frac{|S \cap B(x,r_*)|}{n}} \\ &= \frac{b}{n} + \beta_n^2 + \beta_n \sqrt{\frac{b}{n}} \\ &= \beta_n^2 \left(\frac{100}{\min(\varepsilon,1)^2} + 1 + \frac{10}{\min(\varepsilon,1)} \right) \\ &\leq \frac{p_\varepsilon^2 \min(\varepsilon,1)^2}{111} \frac{111}{\min(\varepsilon,1)^2} = p_\varepsilon. \end{aligned}$$

It follows from Definition 5.4.1 that

$$\left(1+\frac{\varepsilon}{3}\right)^{-1} \frac{p(B(x,r_*))r^k}{r_*^k} \leq p(B(x,r)) \leq \left(1+\frac{\varepsilon}{3}\right) \frac{p(B(x,r_*))r^k}{r_*^k}. \tag{E.20}$$

Finally, by the definition of $Est(x,r,S))$ (Algorithm 5), we have that $Est(x,r,S) = \frac{Est(x,r_*,S)r^k}{r_*^k}$. Combining this with Equation E.20 the definition of $Est(x,r,S)$ (Algorithm 5) along with Equations E.17 and E.18 (which can be safely applied to r_* by reverting to Case 1), we have

$$\begin{aligned} \frac{Est(x,r,S)}{p(B(x,r))} &= \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{p(B(x,r))} \leq \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}\left(1+\frac{\varepsilon}{3}\right)}{\frac{p(B(x,r_*))r^k}{r_*^k}} \\ &= \frac{Est(x,r_*,S)\left(1+\frac{\varepsilon}{3}\right)}{p(B(x,r_*))} \leq \left(1+\frac{\varepsilon}{3}\right)\left(1+\frac{\min(\varepsilon,1)}{9}\right) \\ &\leq 1+\frac{\varepsilon}{2}, \end{aligned}$$

and

$$\begin{aligned} \frac{Est(x,r,S)}{p(B(x,r))} &= \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{p(B(x,r))} \geq \frac{\frac{Est(x,r_*,S)r^k}{r_*^k}}{\frac{p(B(x,r_*))r^k}{r_*^k}\left(1+\frac{\varepsilon}{3}\right)} \\ &= \frac{Est(x,r_*,S)}{p(B(x,r_*))\left(1+\frac{\varepsilon}{3}\right)} \geq \left(1+\frac{\varepsilon}{3}\right)^{-1}\left(1+\frac{\min(\varepsilon,1)}{9}\right)^{-1} \\ &\geq \left(1+\frac{\varepsilon}{2}\right)^{-1}, \end{aligned}$$

which concludes the proof.

□

E.3.4 Proof of Theorem 5.5.1

Theorem E.3.6 (Theorem 5.5.1)**.** *Data_Copy_Detect is a data-copying detector (Definition 5.3.2) with sample complexity at most*

$$m_p(\varepsilon,\delta) = O\left(\frac{d\ln\frac{d}{\delta\varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}\right),$$

for all regular distributions, p.

Proof. Let C be the constant defined in Proposition 5.4.3, and let $n \geq C\frac{d\ln\frac{d}{\delta\varepsilon p_\varepsilon}}{\varepsilon^2 p_\varepsilon}$. Let $S \sim p^n$ be a set of n i.i.d training points, $\{x_1, x_2, \ldots, x_n\}$, and let $q \sim A(S)$ be an arbitrary generated distribution.

By Proposition 5.4.3, the subroutine $Est(x,r,S)$ is accurate over any x and r up to a factor of $(1+\varepsilon)$ with probability at least $1-\frac{\delta}{3}$ (we can achieve this by simply making n a bit larger and substituting $\frac{\delta}{3}$ into Proposition 5.4.3). Suppose this holds, meaning that that for all $x \in \mathbb{R}^d$ and all $r > 0$, the condition of Proposition 5.4.3 holds, and

$$(1+\varepsilon)^{-1} p(B(x,r)) \leq Est(x,r,S) \leq (1+\varepsilon) p(B(x,r)). \tag{E.21}$$

We desire to show that

$$cr_q^{-\varepsilon} - \varepsilon \leq DataCopyDetect(q,S) \leq cr_q^{\varepsilon} + \varepsilon.$$

To do so, we begin applying uniform convergence over $T \sim q^m$. To this end, let

$$\beta_m = \sqrt{\frac{4(d+2)\ln\frac{48m}{\delta}}{m}}.$$

Then by the standard result of Vapnik and Chervonenkis, with probability $1-\frac{\delta}{3}$ over $T\sim q^m$, for all $x\in\mathbb{R}^d$ and all $r>0$,

$$\frac{|T\cap B(x,r)|}{m}-\beta_m\sqrt{\frac{|T\cap B(x,r)|}{m}}\le q(B(x,r))\le\frac{|T\cap B(x,r)|}{m}+\beta_m^2+\beta_n\sqrt{\frac{|T\cap B(x,r)|}{m}}. \tag{E.22}$$

Observe that by the definition of m, we have

$$\begin{aligned}\beta_m^2&=\frac{4(d+2)\ln(48m/\delta)}{m}\\&=4(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\varepsilon^2\min(\varepsilon,1)^2}\ln\left(\frac{98304n^2(d+2)}{\delta\varepsilon^2\min(\varepsilon,1)^2}\right)\right)\frac{\varepsilon^2\min(\varepsilon,1)^2}{2048n^2(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\varepsilon^2\min(\varepsilon,1)^2}\right)}\\&\le 8(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\varepsilon^2\min(\varepsilon,1)^2}\right)\frac{\varepsilon^2\min(\varepsilon,1)^2}{2048n^2(d+2)\ln\left(\frac{98304n^2(d+2)}{\delta\varepsilon^2\min(\varepsilon,1)^2}\right)}\\&=\frac{\varepsilon^2\min(\varepsilon,1)^2}{256n^2}.\end{aligned} \tag{E.23}$$

Next, suppose x,r satisfy that $q(B(x,r))\ge\frac{\varepsilon}{2n}$. For convenience, let $\widehat{q(B(x,r))}$ denote $\frac{|T\cap B(x,r)|}{m}$. By applying Equations E.22 and E.23, it follows that

$$\begin{aligned}\frac{\widehat{q(B(x,r))}}{q(B(x,r))}&\le\frac{q(B(x,r))+\beta_m}{q(B(x,r)}\\&\le 1+\frac{\beta_m}{q(B(x,r)}\\&\le 1+\frac{\min(\varepsilon,1)}{8},\end{aligned}$$

and

$$\begin{aligned}\frac{q(B(x,r))}{\widehat{q(B(x,r))}} &\leq \frac{q(B(x,r))}{q(B(x,r)) - \beta_m^2 - \beta_m\sqrt{\widehat{q(B(x,r))}}} \\ &\leq \frac{q(B(x,r))}{q(B(x,r) - 2\beta_m} \\ &= \frac{1}{1 - \frac{2\beta_m}{q(B(x,r))}} \\ &\leq \frac{1}{1 - \frac{\min(\varepsilon,1)}{4}} \\ &\leq 1 + \frac{\min(\varepsilon,1)}{3}.\end{aligned}$$

Combining these, we have

$$\left(1 + \frac{\min(\varepsilon,1)}{3}\right)^{-1} \leq \frac{q(B(x,r)}{\widehat{q(B(x,r))}} \leq \left(1 + \frac{\min(\varepsilon,1)}{3}\right) \tag{E.24}$$

Next, for $1 \leq i \leq n$, let r_i^* be the radii defined in Algorithm 6. Define $r_i^{-\varepsilon}$ and r_i^{ε} to be the maximal radii r for which q respectively $(\lambda(1+\varepsilon), \gamma(1+\varepsilon)^{-1})$-copies, and $(\lambda(1+\varepsilon)^{-1}, \gamma(1+\varepsilon))$-copies p about x_i. Then we have the following claims.

Claim 1: For $1 \leq i \leq n$, if $q(B(x, r_i^*)) \geq \frac{\varepsilon}{2n}$, $r_i^* \leq r_i^{\varepsilon}$.

Proof. Because $Est(x_i, r_i^*, S) \leq \gamma$, it follows by Equation E.21 that $p(B(x_i, r_i^*)) \leq \left(1 + \frac{\varepsilon}{2}\right)\gamma$. Furthermore, by also applying Equation E.24 we have that

$$\frac{q(x_i, r_i^*)}{p(x_i, r_i^*)} \geq \frac{\frac{|B(x_i, r_i^*) \cap T|}{m}}{Est(x_i, r_i^*, S)\left(1 + \frac{\min(\varepsilon,1)}{3}\right)\left(1 + \frac{\varepsilon}{2}\right)} \geq \lambda(1+\varepsilon)^{-1}.$$

Thus q $(\lambda(1+\varepsilon)^{-1}, \gamma(1+\varepsilon))$-copies all points in $B(x_i, r_i^*)$ implying $r_i^* \leq r_i^{\varepsilon}$. □

Claim 2: For $1 \leq i \leq n$, if $q(B(x, r_i^{-\varepsilon})) \geq \frac{\varepsilon}{2n}$, then $r_i^{-\varepsilon} \leq r_i^*$.

Proof. For the left hand side, we use a similar argument. By Equation E.21 along with the

definition of r_i^{ε}, we have $Est(x_i, r_i^{-\varepsilon}, S) \leq \gamma(1+\varepsilon)^{-1}\left(1+\frac{\varepsilon}{2}\right) \leq \gamma$. By Equations E.21 and E.24, we have

$$\frac{\frac{|B(x_i, r_i^{-\varepsilon}) \cap T|}{m}}{Est(x_i, r_i^*, S)} \geq \frac{q(B(x_i, r_i^{-\varepsilon}))}{p(B(x_i, r_i^{-\varepsilon}))\left(1+\frac{\min(\varepsilon,1)}{3}\right)\left(1+\frac{\varepsilon}{2}\right)} \geq \lambda,$$

with the last inequality coming again from the definition of $r_i^{-\varepsilon}$. Thus, $r_i^{-\varepsilon}$ meets the criteria from Algorithm 6 required to be selected as r_i^*. As a technical note, because Algorithm 6 only considers finitely many radii, it may not consider precisely $r_i^{-\varepsilon}$. However, this is not a problem, as the nearest considered radii to this point have nearly unchanged values of $Est(x, r, S)$ and $\frac{|B(x,r)\cap T|}{m}$, meaning that some similar radius will be considered. □

Finally, armed with our claims, we now consider the total region of points in which Algorithm 6 claimed data-copying occurs. Let S^1 and S^2 be the sets of indices for which the conditions are violated for claims 1 and 2 respectively. Then it follows from Claim 1 that

$$\begin{aligned} cr_q^{\varepsilon} - q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) &= q\left(\cup_{i=1}^n B(x_i, r_i^{\varepsilon})\right) - q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) \\ &\geq q\left(\cup_{i=1}^n B(x_i, r_i^{\varepsilon})\right) - q\left(\cup_{i \notin S^1} B(x_i, r_i^*)\right) - q\left(\cup_{i \in S^1} B(x_i, r_i^*)\right) \\ &\geq -\frac{\varepsilon}{2}. \end{aligned}$$

Here we are using Claim 1 to hand all terms that are not in S^1, and then crudely bounding the remaining terms with $\frac{\varepsilon}{2n}$. Similarly, by Claim 2, we have

$$\begin{aligned} q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - cr_q^{-\varepsilon} &= q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - q\left(\cup_{i=1}^n B(x_i, r_i^{-\varepsilon})\right) \\ &\geq q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) - q\left(\cup_{i \notin S^2} B(x_i, r_i^{-\varepsilon})\right) - q\left(\cup_{i \in S^2} B(x_i, r_i^{-\varepsilon})\right) \\ &\geq -\frac{\varepsilon}{2}. \end{aligned}$$

Combining these, we see that

$$cr_q^{-\varepsilon} - \frac{\varepsilon}{2} \leq q\left(\cup_{i=1}^n B(x_i, r_i^*)\right) \leq cr_q^{\varepsilon} + \frac{\varepsilon}{2}.$$

All the remains is to show that our last step of Algorithm 6, in which we estimate this mass, is accurate up to a factor of $\frac{\varepsilon}{2}$. However, this immediately follows from the fact that we use $\frac{20\log\frac{1}{\delta}}{\varepsilon^2}$ samples (last line of Algorithm 6). In particular, because this holds with probability $1-\frac{\delta}{3}$, we can apply a union bound with our other two probabilistic events (*Est* being sufficiently close, and T yielding uniform convergence) to get a total failure probability of δ, as desired. □

E.3.5 Proof of Theorem 5.6.1

Theorem E.3.7 (Theorem 5.6.1)**.** *Let B be a data-copying detector. Let $\varepsilon = \delta = \frac{1}{3}$. Then there exist 1-regular distributions p for which p_ε is arbitrarily small and B has sample complexity*

$$m_p(\varepsilon,\delta) \geq \Omega(\frac{1}{p_\varepsilon}).$$

More precisely, for all integers $\kappa > 0$, there exists a probability distribution p such that $\frac{1}{9\kappa} \leq p_\varepsilon \leq \frac{1}{\kappa}$, and $m_p(\varepsilon,\delta) > \Omega(\kappa)$.

Proof Outline: Let κ be a sufficiently large integer. Then we take the following steps.

1. We define the probability distribution p_T, where $T \subset [2\kappa] = \{1,2,\ldots,2\kappa\}$ is a subset with $|T| = \kappa$ that parametrizes our distribution. We then show that for all T, p_T is a 1-regular distribution satisfying $\frac{1}{9\kappa} \leq (p_T)_\varepsilon \leq \frac{1}{\kappa}$.

2. We define a generative algorithms A_T and A'_T, where as before $T \subset [2\kappa]$ with $|T| = \kappa$. We then show that if $S \sim p_T^{O(\kappa)}$, $A_T(S)$ is likely to have a high data-copy rate with respect to p_T, whereas $A'_T(S)$ has a data-copy rate of 0.

3. We construct families

$$\mathscr{F} = \{(p_T, A_T) : T \subset [2\kappa], |T| = \kappa\} \text{ and } \mathscr{F}' = \{(p_T, A'_T) : T \subset [2\kappa], |T| = \kappa\},$$

and show that $(S,A(S))$ follows very similar distributions when S is drawn from $p^{O(\kappa)}$

and (p,A) is drawn from $\mathscr{F}$ and $\mathscr{F}'$ respectively, meaning that it is difficult to tell which family the pair (p,A) is drawn from.

4. We show that if B has sample complexity at most $O(\kappa)$, then by (2.) it would be able to distinguish $(S,A_T(S))$ from $(S,A'_T(S))$ thus contradicting (3.) We thus conclude B has sample complexity $\Omega(\kappa)$, as desired.

Proof. We follow the outline above proceeding step by step.

Step 1: constructing p_T

First, set $\gamma < 1$ arbitrarily, and let $\lambda = 13$. Note that these constants are chosen out of convenience, and for different values of ε, δ, different ones can be chosen.

Let $\kappa > 0$ be any integer, and let $[2\kappa] = \{1,2,3,\ldots,2\kappa\}$. Let $C_1, C_2, \ldots, C_{2\kappa}$ be 2κ disjoint unit circles in $\mathbb{R}^d$ with distance at least 3 between any two circles. All data distributions, p_T, that we construct will have support over $\cup_{i=1}^{2\kappa} C_i$, and will further obey the constraint that their marginal distribution over any C_i is precisely the uniform distribution. Thus, a distribution p_T is uniquely specified by the probability mass it assigns to each circle. To this end, we define p_T as follows.

Definition E.3.8. *Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$. Then p_T is the unique probability distribution satisfying the criteria above such that*

$$p_T(C_i) = \begin{cases} \frac{1}{3\kappa} & i \in T \\ \frac{2}{3\kappa} & i \notin T \end{cases}$$

Lemma E.3.9. *p_T is 1-regular, and satisfies $\frac{1}{9\kappa} \leq (p_T)_\varepsilon \leq \frac{2}{3\kappa}$ when $\varepsilon = \frac{1}{3}$.*

Proof. First, we observe that by Proposition 5.4.2, we immediately have that p_T is 1-regular as a union of disjoint circles is a 1 dimensional closed manifold, and the density function of p_T with respect to each circle is uniform and therefore continuous. For convenience, we let p denote p_T, as by symmetry, $(p_T)_\varepsilon$ is equal for all values of T.

Next, for $r \leq 2$ and $x \sim p$, we compute $\frac{p(B(x,r)}{r}$. Suppose $x \in C_i$. The key observation is that the density of p over C_i is uniform, and thus since $r \leq 2$, the mass of $B(x,r)$ can be found by simply computing the arc length. It follows that

$$\frac{p(B(x,r))}{r} = p(C_i))\frac{4\arcsin(\frac{r}{2})}{2\pi r}. \tag{E.25}$$

By some basic properties about arcsin, it follows that $\frac{p(B(x,r)}{r}$ is monotonically increasing with $0 < r \leq 2$ and satisfies $\lim_{r\to 0^+} \frac{p(B(x,r)}{r} = \frac{p(C_i)}{\pi}$ and $\frac{p(B(x,2))}{2} = \frac{p(C_i)}{2}$. Using this, we now prove the upper and lower bounds for p_ε beginning with the upper bound.

Assume towards a contradiction that $p_\varepsilon > \frac{2}{3\kappa}$. By Definition 5.4.1, this implies that for any sufficiently small $r > 0$, we have

$$\left(1+\frac{\varepsilon}{3}\right)^{-1}\frac{p(B(x,r))}{r} \leq \frac{p(B(x,2))}{2} \leq \left(1+\frac{\varepsilon}{3}\right)\frac{p(B(x,r))}{r},$$

as for any $x \sim p$, $p(B(x,2)$ is at most $\frac{2}{3\kappa}$. Substituting equation E.25 and taking the limit as $r \to 0^+$, it follows that $\frac{p(C_i)}{2} \leq \frac{7}{6}\frac{p(C_i)}{\pi}$, which is a contradiction giving us that $p_\varepsilon \leq \frac{2}{3\kappa}$.

Next, for the lower bound, it suffices to show that for any x and any $0 < s \leq r$ with $p(B(x,r)) \leq \frac{1}{9\kappa}$ that

$$\left(1+\frac{\varepsilon}{3}\right)^{-1}\frac{p(B(x,s))}{s} \leq \frac{p(B(x,r))}{r} \leq \left(1+\frac{\varepsilon}{3}\right)\frac{p(B(x,s))}{s}. \tag{E.26}$$

Applying Equation E.25 with $r = 1$, we have for any $x \sim p$,

$$\frac{p(B(x,1))}{1} = p(C_i)\frac{4\arcsin(\frac{1}{2})}{2\pi} = p(C_i)\frac{1}{3} \geq \frac{1}{3\kappa}\frac{1}{3} = \frac{1}{9\kappa}.$$

Since $\frac{p(B(x,r)}{r}$ is monotonic in r, it follows that $p(B(x,r)) \leq \frac{1}{9\kappa}$ only if $r \leq 1$. We are now prepared to prove Equation E.26.

The left inequality immediately holds since $\frac{p(B(x,r)}{r}$ is monotonic in r. For the right

inequality, we have that if r satisfies $p(B(x,r) \leq \frac{1}{9\kappa}$, then $r \leq 1$ implying for $x \in C_i$,

$$\begin{aligned}
\frac{p(B(x,r))}{r} &\leq \frac{p(B(x,1)}{1} \\
&= p(C_i)\frac{1}{3} \\
&\leq (1+\frac{1}{9})\frac{p(C_i)}{\pi} \\
&= \left(1+\frac{\varepsilon}{3}\right)\lim_{t\to 0}\frac{p(B(x,t))}{t} \\
&\leq \left(1+\frac{\varepsilon}{3}\right)\frac{p(B(x,s))}{s},
\end{aligned}$$

as desired. □

Step 2: defining A_T and A'_T

Having defined our probability distributions, p_T, we now define our generative algorithms A_T and A'_T. Recall that a generative algorithm, A, is any process that takes as input a set of points $S \in \mathbb{R}^d$ and then returns a probability distribution, $A(S)$ over $\mathbb{R}^d$. The algorithm is allowed to have randomization.

A_T and A'_T will always be constrained to output distributions that are similar to p_T in the sense that they have support over a disjoint union of circles, and their marginal distribution over any circle (within the support) is the uniform distribution. The only change is that we add one extra circle, C_0, that satisfies

$$||C_0 - C_i|| \geq 2 + \max_{i,j} ||C_i - C_j||,$$

meaning that it is very far from all C_i. Thus, any outputted distribution by A_T or A'_T can be specified by specifying the probability mass it assigns to each circle in $\{C_0, C_1, \ldots, C_{2\kappa}\}$.

Both A_T and A'_T will operate under the assumption that the training sample of points S is relatively well behaved. In the event that this does not hold, A_T and A'_T will output the uniform distribution over C_0 as a default. We now formally define this criteria upon S.

Definition E.3.10. *Let S be a finite set of points and $T \subset [2\kappa]$ be a set of indices with $|T| = \kappa$. We say that S covers T the sets $L = \{i : i \in T, |C_i \cap S| = 1\}$ and $L' = \{i : i \notin T, |C_i \cap S| = 1\}$ both satisfy $|L|, |L'| \geq \frac{\kappa}{8}$.*

Observe that this definition if symmetric with respect to complements meaning that S covers T if and only if S covers $[2\kappa] \setminus T$. We now use this to define A_T and A'_T beginning with A_T.

Definition E.3.11. *Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$, and let S be any set of points in $\mathbb{R}^d$. Then A_T consists of the following steps. We let q denote its output, and $A_T(S)$ denote the full distribution of potential generated distributions q.*

1. *If S does not cover T, then output the uniform distribution over C_0 as q.*
2. *Otherwise, let $L = \{i : i \in T, |C_i \cap S| = 1\}$ be as defined in Definition E.3.10.*
3. *Randomly select $L_* \subset L$ with $|L_*| = \frac{\kappa}{8}$ at uniform.*
4. *We then let q be the unique probability distribution satisfying the criteria above with*

$$q(C_i) = \begin{cases} \frac{\lambda(1+\varepsilon)}{3\kappa} & i \in L_* \\ 0 & i \in [2\kappa] \setminus L_* \\ 1 - \frac{\lambda(1+\varepsilon)}{24} & i = 0 \end{cases}$$

Having defined A_T, we define A'_T by having $A'_T = A_{[2\kappa]\setminus T}$. That is,

Definition E.3.12. *Let $T \subset [2\kappa]$ be a subset of indices with $|T| = \kappa$, and let S be any set of points in $\mathbb{R}^d$. Then $A'_T(S)$ is precisely $A_{[2\kappa]\setminus T}(S)$ where $[2\kappa] \setminus T$ is the complement of T.*

Observe that if S covers T, then by Definitions E.3.11 and E.3.12, $A_T(S)$ and $A'_T(S)$ will both have supports non-trivially intersecting the set of circles over which p_T is based, $\cup_{i=1}^{2\kappa} C_i$. We now show that this condition is sufficient for our desired behavior with respect to data-copying.

Lemma E.3.13. *Let κ satisfy $\frac{1}{3\kappa} \leq \gamma$. For any $T \subset [2\kappa]$, let S be any set of points in the support of p_T that covers T. Then with probability 1 over the randomness of A_T and A'_T, $q_T \sim A_T(S)$ and $q'_T \sim A'_T(S)$ have respective data-copy rates $cr^{-\varepsilon}_{q_T}$ and $cr^{\varepsilon}_{q'_T}$ satisfying*

$$cr^{-\varepsilon}_{q_T} \geq \frac{\lambda(1+\varepsilon)}{24},$$

$$cr^{\varepsilon}_{q'_T} = 0.$$

Proof. Let L and L' be as in Definition E.3.10. We begin with $cr^{-\varepsilon}_{q_T}$, which was the data-copy rate of q_T with parameters $(\lambda(1+\varepsilon), \gamma(1-\varepsilon))$ (Definition 5.3.3).

Since $|L| \geq \frac{\kappa}{8}$, there exists $L_* \subset L$ with $|L_*| = \frac{\kappa}{8}$ such that q_T has support over $C_0 \cup \{C_i\}_{i \in L_*}$. For any $i \in L_*$, let x_i denote the unique point in the intersection of C_i and S. Observe that by the definition of L, $p_T(B(x_i, 2)) = \frac{1}{3\kappa}$. On the other hand, we have $q_T(B(x_i, 2)) = q_T(C_i) = \frac{\lambda(1+\varepsilon)}{3\kappa}$, with the first equality holding since C_i is the only circle that intersects $B(x_i, 2)$. It follows by Definition 5.2.1 that q_T $(\lambda(1+\varepsilon), \gamma(1+\varepsilon)^{-1})$-copies all $x \in C_i$. Taking the total measure (under q_T), we have

$$cr^{-\varepsilon}_{q_T} \geq q_T(\cup_{i \in L_*} C_i) = \frac{\kappa}{8} \frac{\lambda(1+\varepsilon)}{3\kappa} = \frac{\lambda(1+\varepsilon)}{24\kappa}$$

as desired.

Next, we show $cr^{\varepsilon}_{q'_T} = 0$. To do so, it suffices to show that for all $x \in S$ and $r > 0$,

$$q'_T(B(x, r)) < \lambda(1+\varepsilon)^{-1} p_T(B(x, r)),$$

as this would imply that no points are $(\lambda(1+\varepsilon)^{-1}, \gamma(1+\varepsilon))$-copied.

Observe that $M = \cup_{1 \leq i \leq 2\kappa} C_i$ is a 1-dimensional manifold containing the entire support of p_T, and that furthermore the marginal distribution of $q'_T(S)$ over M has a well defined probability density with respect to M. Since $x \in S$ and $S \subset M$ (as $S \subset supp(p_T)$), we can consider two cases:

if $B(x,r)$ intersects C_0 (the only region in the support of $A'_T(S)$ outside M), and if $B(x,r)$ does not intersect C_0.

Case 1: $B(x,r)$ **intersects** C_0

Observe that by the definition of C_0, $C_i \subset B(x,r)$ for all $1 \leq i \leq 2\kappa$. This is because C_0 is very far from all the other circles. However, this implies $M \subset B(x,r)$ meaning that $p_T(B(x,r)) \geq p_T(M) = 1$. However, $q'_T(B(x,r))$ is clearly at most 1, making the desired inequality trivially hold as $\lambda(1+\varepsilon)^{-1} > 1$.

Case 2: $B(x,r)$ **does not intersect** C_0

Observe that this implies $supp(p_T) \cap B(x,r) = supp(q'_T \cap B(x,r) \subseteq M$, as both of these distributions only have support on M when outside of C_0. Since p_T and q'_T both have well defined probability densities over M, their masses over $B(x,r)$ can be found by integrating their densities over this region.

However, by the definition of A'_T, for any $y \in supp(q'_T)$, we have that $y \in C_i$ where $i \in [2\kappa] \setminus T$. By letting p_T and q'_T denote their respective density functions, it follows that

$$p_T(y) = \frac{2}{3\kappa(2\pi)}, \text{ and } q'_T(y) = \frac{\lambda(1+\varepsilon)}{3\kappa(2\pi)}.$$

It follows that $\frac{q'_T(y)}{p_T(x)} = \frac{\lambda(1+\varepsilon)}{2} < \lambda(1+\varepsilon)^{-1}$. Thus, it follows from integrating as y goes over $B(x,r)$ that $q'_T(B(x,r)) < \lambda(1+\varepsilon)^{-1} p_T(B(x,r))$ as desired.

As a slight technical detail, while this inequality will no longer be strict if $p_T(B(x,r)) = 0$, we know that this is never the case since $p_T(B(x,r))$ is strictly positive for all $x \in M$.

□

Next, we bound the probability that set of κ points drawn i.i.d. from p_T, $S \sim p_T^{\kappa}$, will cover T. To do so, we begin with a combinatorial lemma.

Lemma E.3.14. *Let m,n be an integers with $\frac{n}{4} \leq m \leq \frac{3n}{4}$. Suppose m numbers are chosen uniformly at random from $\{1,2,\ldots,n\}$. Then with probability at least $1 - 2\exp\left(\frac{-n}{2048}\right)$, at least*

$\frac{n}{8}$ *numbers in* $\{1,2,\ldots,n\}$ *are selected exactly once.*

Proof. Let $b_1, b_2, \ldots, b_m$ denote our m numbers chosen from $\{1,2,\ldots,n\}$. For $1 \leq i \leq m$, let X_i be an indicator variable for b_i being distinct from x_j for all $1 \leq j < i$, and let $Y_i = 1 - X_i$ be an indicator variable for the opposite. By convention we take $X_1 = 1$ and $Y_1 = 0$. Let $X = \sum_{i=1}^{m} X_i$ and $Y = \sum_{i=1}^{m} Y_i$. The key observation is that if Z denotes the number of elements in $\{1,\ldots,n\}$ that are selected exactly once, then $Z \geq X - Y$.

To see this, observe that if we maintain Z as a set while observing $b_1, b_2, \ldots, b_m$, then it follows that whenever $X_i = 1$, we append an element to Z (as its corresponding number b_i will have occurred for the first time and thus be chosen exactly once), and we remove an element from Z only when $Y_i = 1$, as a repeat of a number necessarily implies $Y_i = 1$. It follows that to bound Z, it suffices to bound $X - Y$.

To this end, observe that for any $1 \leq i \leq m$, *regardless* of the outcomes of $X_1, X_2, \ldots, X_{i-1}$, $\mathbb{E}[X_i] \geq \frac{n-i+1}{n}$, as there are at least $n - i + 1$ numbers in $\{1,\ldots,n\}$ that have not been chosen yet. It follows that if $X_i^* = \sum_{j=1}^{i} X_i - \frac{n-i+1}{n}$ for $1 \leq i \leq m$, then X_i^* is a sub-martingale (as each term in the sum has expected value at least 0) satisfying $|X_i^* - X_{i-1}^*| \leq 1$. Applying Azuma's inequality, we see that

$$\Pr[X_m^* \geq -\frac{n}{32}] \geq 1 - \exp\left(\frac{-n^2}{2048m}\right) \geq 1 - \exp\left(\frac{-n}{2048}\right).$$

We now apply a similar trick for $Y_1, \ldots, Y_m$. In this case, observe that for $1 \leq i \leq m$, *regardless* of the outcomes of $Y_1, \ldots, Y_{i-1}$, $\mathbb{E}[Y_i] \leq \frac{i-1}{m}$, as there can be at most $i - 1$ numbers that have already been chosen and $Y_i = 1$ if and only if the corresponding b_i is equal to one of those $i - 1$ numbers. It follows that $Y_i^* = \sum_{j=1}^{i} Y_i - \frac{i-1}{m}$ is a super-martingale (as each term has expected value at most 0) with $|Y_i^* - Y_{i-1}^*| \leq 1$. Applying Azuma's inequality, we see that

$$\Pr[Y_m^* \leq \frac{n}{32}] \geq 1 - \exp\left(\frac{-n^2}{2048m}\right) \geq 1 - \exp\left(\frac{-n}{2048}\right).$$

Applying a union bound, we see that with probability at least $1-2\exp\left(\frac{-n}{2048}\right)$, $X_m^* \geq \frac{-n}{32}$ and $Y_m^* \leq \frac{-n}{32}$. By substituting these inequalities in, it follows that with probability $1-2\exp\left(\frac{-n}{2048}\right)$, Z satisfies

$$\begin{aligned}
Z &\geq X - Y \\
&= \sum_{i=1}^{m} X_i - \sum_{j=1}^{m} Y_i \\
&= \sum_{i=1}^{m} \left(X_i - \frac{n-i+1}{n} \right) + \sum_{i=1}^{m} \left(\frac{n-i+1}{n} \right) - \sum_{i=1}^{m} \left(Y_i - \frac{i-1}{n} \right) - \sum_{i=1}^{m} \left(\frac{i-1}{n} \right) \\
&= X_m^* - Y_m^* + \sum_{i=1}^{m} \left(\frac{n-i+1}{n} \right) - \sum_{i=1}^{m} \left(\frac{i-1}{n} \right) \\
&\geq -\frac{n}{32} - \frac{n}{32} + \sum_{i=1}^{m} \left(\frac{n-i+1}{n} \right) - \sum_{i=1}^{m} \left(\frac{i-1}{n} \right) \\
&= -\frac{n}{16} + m \left(\frac{n+(n-m+1)}{2n} \right) - \frac{m(m-1)}{2n} \\
&= -\frac{n}{16} + \frac{m}{2n} (2n - m + 1 - m + 1) \\
&= -\frac{n}{16} + \frac{m(n-m+1)}{n} \\
&\geq -\frac{n}{16} + \frac{3n}{16} = \frac{n}{8},
\end{aligned}$$

with the last inequality holding since $\frac{n}{4} \leq m \leq \frac{3n}{4}$. This concludes our proof since we have shown $Z \geq \frac{n}{8}$ with the desired probability.

□

We now apply Lemma E.3.14 to bound the probability that $S \sim p_T^\kappa$ covers T.

Lemma E.3.15. *Let $T \subset [2\kappa]$ be a set of κ indices, and let $S \sim p_T^\kappa$ be a set of κ i.i.d points . Then with probability at least $1-4\exp\left(-\frac{\kappa}{2048}\right)$, S covers T.*

Proof. Let $S = (x_1, x_2, \ldots, x_\kappa)$, and let $A = (a_1, a_2, \ldots, a_\kappa)$ be the unique indices such that $x_i \in a_i$. By Definition E.3.10, L and L' are the number of values in T and $[2\kappa] \setminus T$ that appear exactly once in A. We desire to bound the probability that $|L| \geq \frac{\kappa}{8}$ and $|L'| \geq \frac{\kappa}{8}$. To do so, the key idea is

to condition on M, which we define as the number of $1 \leq i \leq \kappa$ such that $a_i \in T$.

Suppose that $M = m$. Observe that the conditional distribution of A (viewed as a multiset) given $M = m$ is precisely the distribution obtained by selecting m indices at uniform from T and $\kappa - m$ indices at uniform from $[2\kappa] \setminus m$. This holds because p_T is uniform when restricted to $\cup_{i \in T} C_i$ or $\cup_{i \in [2\kappa] \setminus T} C_i$. Suppose that $\frac{\kappa}{4} \leq m \leq \frac{3\kappa}{4}$. Then the same must hold for $\kappa - m$. it follows by applying Lemma E.3.14 to selecting m indices from T and $\kappa - m$ indices from $[2\kappa] \setminus T$ that with probability at least $1 - 2\exp\left(-\frac{\kappa}{2048}\right)$ that $|L| \geq \frac{\kappa}{8}$ and $|L|' \geq \frac{\kappa}{8}$. Thus, by summing over all such m, we see that

$$\begin{aligned}
\Pr_{S \sim p_T^\kappa}[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8}] &= \sum_{m=1}^{\kappa} \Pr_{S \sim p_T^\kappa}(M = m) \Pr[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8} | M = m] \\
&\geq \sum_{m=\kappa/4}^{3\kappa/4} \Pr_{S \sim p_T^\kappa}(M = m) \Pr[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8} | M = m] \\
&\geq \sum_{m=\kappa/4}^{3\kappa/4} \Pr_{S \sim p_T^\kappa}(M = m) \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right) \\
&= \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right) \Pr_{S \sim p_T^\kappa}[\frac{\kappa}{4} \leq M \leq \frac{3\kappa}{4}].
\end{aligned}$$

To bound the latter probability, we simply apply a Chernoff bound, as $M = \sum_{i=1}^{\kappa} \mathbb{1}(a_i \in T)$ is the sum of κ independent indicator variables each with expected value $\frac{1}{3}$. Using a two sided Chernoff bound, we see that $\Pr[\frac{\kappa}{4} \leq M \leq \frac{3\kappa}{4}] \geq 1 - 2\exp\left(-\frac{\kappa}{144}\right)$. Substituting this, it follows that

$$\Pr_{S \sim p_i^\kappa}[|L| \geq \frac{\kappa}{8}, |L'| \geq \frac{\kappa}{8}] \geq \left(1 - 2\exp\left(-\frac{\kappa}{2048}\right)\right)\left(1 - 2\exp\left(-\frac{\kappa}{144}\right)\right) \geq 1 - 4\exp\left(-\frac{\kappa}{2048}\right).$$

□

Step 3: Constructing $\mathscr{F}$ and $\mathscr{F}'$

We start by defining $\mathscr{F}$ and $\mathscr{F}'$ as distributions of pairs (p, A) where p is a data distribution and A is a generative algorithm.

Definition E.3.16. *$\mathscr{F}$ and $\mathscr{F}'$ are the uniform distributions over $\{(p_T, A_T) : T \subset [2\kappa], |T| = \kappa\}$ and $\{(p_T, A'_T) : T \subset [2\kappa], |T| = \kappa\}$ respectively.*

Next, we use $\mathscr{F}$ and $\mathscr{F}'$ to construct distributions Q and Q' over pairs (S, q), where S is a set of points, and q is generated distribution.

Definition E.3.17. *Let Q be the distribution of (S, q) where $(p_T, A_T) \sim \mathscr{F}$, $S \sim p_T^\kappa$, and $q \sim A_T(S)$. Similarly, let Q' be the distribution of (S, q) where $(p_T, A'_T) \sim \mathscr{F}'$, $S \sim p_T^\kappa$, and $q \sim A'_T(S)$.*

Our goal will be to show that Q and Q' follow similar distributions. Our strategy will be to show that for the majority of (S, q) in their supports, they have similar probability masses. To this end, we first characterize the values of (S, q) that we are interested in considering.

Definition E.3.18. *We say that (S, q) is nice if S is a sample of points from some p_T, and q is a generated distribution from either A_T or A'_T that has no support over C_0. More precisely, (S, q) is nice if the following conditions hold:*

1. *$S \subset \cup_{i=1}^{2\kappa} C_i$, with $|S| = \kappa$.*
2. *There exists a set of $\frac{\kappa}{8}$ distinct indices, $L_* \subset [2\kappa]$, such that for $0 \leq i \leq 2\kappa$,*

$$q(C_i) = \begin{cases} \frac{\lambda(1+\varepsilon)}{3\kappa} & i \in L_* \\ 0 & i \in [2\kappa] \setminus L_* \\ 1 - \frac{\lambda(1+\varepsilon)}{24} & i = 0 \end{cases}$$

3. *For every $i \in L_*$, $|S \cap C_i| = 1$, meaning exactly one element from S is in C_i.*

We now prove a quick lemma relating nice pairs to instances in which S covers T.

Lemma E.3.19. *Let $T \subset [2\kappa]$ satisfy $|T| = \kappa$. Let $S \sim p_T^\kappa$ and let q and q' be generated distributions with $q = A_T(S)$ and $q' = A'_T(S)$. Then the following three are equivalent:*

1. *(S, q) is nice.*

2. (S, q') *is nice.*

3. *S covers T.*

Proof. Suppose S covers T. Then the sets L and L' (Definition E.3.10) each have size at least $\frac{\kappa}{8}$ implying that when running A_T or A'_T, the set L_* will be non-trivial. This in turn will imply that (q,S) and (q',S) are nice, regardless of the choice of L_*.

Otherwise, suppose S does not cover T. Then by Definition E.3.11, $A_T(S)$ and $A_{T'}(S)$ will both be the uniform distribution over C_0 thus violating Definition E.3.18. □

We now show that Q and Q' assign identical probability masses to nice pairs.

Lemma E.3.20. *Let (S,q) be a nice pair. Then $Q(S,q) = Q'(S,q)$ with these expressions denoting the probability that (S,q) is chosen over Q and Q' respectively.*

Proof. Let $S = \{x_1, x_2, \ldots, x_\kappa\}$. Let M denote the set of indices in $\{1,2\ldots,2\kappa\}$ such that exactly one point of S lies in the corresponding circle. That is, $M = \{i : |S \cap C_i| = 1\}$. Let L_* be the set of indices in $\{1,2,\ldots 2\kappa\}$ where q assigns non-trivial probability mass to the corresponding circle. That is, $L_* = \{i : q(C_i) > 0, 1 \le i \le 2\kappa\}$. Since (S,q) is a nice pair (Definition E.3.18), L_* is a subset of M, and satisfies $|L_*| = \frac{\kappa}{8}$. Furthermore, q is uniquely determined by L_*.

We now compute $Q(S,q)$ and $Q'(S,q)$ by summing the conditional probabilities of (S,q) given (p_T, A_T) and (p_T, A'_T) respectively as T ranges over all subsets. By utilizing the fact that (S,q) is nice (meaning it can only occur if S covers T) along with the definition of A_T, we have that

$$\begin{aligned}
Q(S,q) &= \sum_{|T|=\kappa : T \subset [2\kappa]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[(S,q) | p_T, A_T] \\
&= \sum_{|T|=\kappa : T \subset [2\kappa]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \Pr[A_T(S) = q | S, T] \\
&= \sum_{T : S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \Pr[A_T(S) = q | S, T]. \\
&= \sum_{T : S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}(L_* \subseteq T)}{\binom{|T \cap M|}{\kappa/8}}.
\end{aligned}$$

with the last equality holding because $A_T(S)$ randomly chooses a $\kappa/8$ element subset of $T \cap M$ for the support of q (see Definition E.3.11). The term $\mathbb{1}(L_* \subseteq T)$ is necessary because if $L_* \not\subseteq T$, then it is impossible for it to be chosen making the probability 0.

Similarly, letting T^c denote the complement of T, we have

$$Q'(S,q) = \sum_{T:S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}\left(L_* \subseteq T^c\right)}{\binom{|T^c \cap M|}{\kappa/8}},$$

with the only real difference being the support is chosen from $T^c \cap M$ rather than $T \cap M$.

To show that these sums are equal, we will further group the sums by using M to define an equivalence relation over $\{T : T \subset [2\kappa], |T| = \kappa\}$. For $T_1, T_2 \subset [2\kappa]$, we say they are equivalent if their intersections with $[2\kappa] \setminus M$, the complement of M, are equal. That is,

$$T_1 \sim T_2 \iff T_1 \cap ([2\kappa] \setminus M) = T_2 \cap ([2\kappa] \setminus M).$$

The usefulness of this equivalence relation is in the following claim.

Claim: Let $T_1 \sim T_2$ be equivalent subsets of κ indices. Then the following hold:

1. $\Pr[S|p_{T_1}] = \Pr[S|p_{T_2}]$.
2. $|T_1 \cap M| = |T_2 \cap M|$ and $T_1^c \cap M| = |T_2^c \cap M|$.
3. S covers T_1 if and only if S covers T_2.

Proof. (Of Claim) Let T be any set of indices, let $S = \{x_1, x_2, \ldots, x_\kappa\}$, and let $a_1, a_2, \ldots a_\kappa$ denote the respective indices of the circles that $x_1, \ldots, x_\kappa$ are on. Without loss of generality (relabeling if necessary), suppose that $a_1, a_2, \ldots, a_m$ are the unique indices that constitute M (defined above).

Since p_T has probability mass $\frac{1}{3\kappa}$ on every index in T and $\frac{2}{3\kappa}$ on the others, we have that

the probability density of S (denoted $\Pr[S|p_{T_1}]$) satisfies,

$$\begin{aligned}\Pr[S|p_T] &= \prod_{i=1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T)}{\kappa}\frac{1}{2\pi} \\ &= \left(\prod_{i=1}^{m} \frac{2-\mathbb{1}(a_i \in T \cap M)}{\kappa}\frac{1}{2\pi}\right)\left(\prod_{i=m+1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T \cap ([2\kappa]\setminus M))}{\kappa}\frac{1}{2\pi}\right) \\ &= \left(\frac{2^{|T^c \cap M|}}{(2\pi\kappa)^m}\right)\left(\prod_{i=m+1}^{\kappa} \frac{2-\mathbb{1}(a_i \in T \cap ([2\kappa]\setminus M))}{\kappa}\frac{1}{2\pi}\right),\end{aligned}$$

with the last equality exploiting the fact that $\{a_1, a_2, \ldots, a_m\}$ precisely equals M (by the definition of M). Next, observe that if $T_1 \sim T_2$, then by definition, $T_1 \cap [2\kappa] \setminus M = T_2 \cap [2\kappa \setminus M$ implying that the second part of the product is equal. However, since $|T_1| = |T_2| = \kappa$, the first part must be equal as well, as $|T^c \cap M| = \kappa - |T^c \cap [2\kappa] \setminus M|$. It follows that the probability densities are the same. Note that this observation also implies the second claim, that $|T_1 \cap M| = |T_2 \cap M|$ and $T_1^c \cap M| = |T_2^c \cap M|$

Finally, to show the second part of the claim, we simply observe that for a set T, the sets L and L' from Definition E.3.10 are precisely $T \cap M$ And $T^c \cap M$. For $T = T_1, T_2$, by the second claim, these have equal sizes. □

We now return to the proof of Lemma E.3.20. Having shown the claim, we now return to our original computation. Let $T_1, T_2, \ldots, T_r$ denote sets of κ indices with $[T_1], [T_2], \ldots, [T_r]$ denoting their respective equivalence classes such that $[T_1], \ldots, [T_r]$ partition $\{T : S \text{ covers } T\}$. This is possible from the third part of our claim.

For $1 \leq i \leq r$, let $m_i = |T_i \cap M|$ and $m_i' = |T_i^c \cap M|$ where T_i^c denotes the complement of T_i. It follows from second part of our claim that $|T \cap M|, |T^c \cap M|$ both equal m_i as well for all $T \in [T_i]$.

By partitioning our sum for $Q(S,q)$ in using $[T_1],\ldots,[T_r]$, we have

$$\begin{aligned}
Q(S,q) &= \sum_{T:S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}(L_* \subseteq T)}{\binom{|T\cap M|}{\kappa/8}} \\
&= \sum_{i=1}^{r} \sum_{T\in[T_i]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}(L_* \subseteq T)}{\binom{|T\cap M|}{\kappa/8}} \\
&= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \sum_{T\in[T_i]} \frac{\mathbb{1}(L_* \subseteq T)}{\binom{m_i}{\kappa/8}} \\
&= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}},
\end{aligned}$$

with the last equality coming by counting the number of $T \in [T_i]$ such that $L_* \subseteq T$. This counting problem essentially forces all $\kappa/8$ elements in L_* to be in T leaving us to choose the remaining elements in M that can be part of T.

By using the exact same line of reasoning for $Q'(S,q)$, we have

$$\begin{aligned}
Q'(S,q) &= \sum_{T:S \text{ covers } T} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}(L_* \subseteq T^c)}{\binom{|T^c\cap M|}{\kappa/8}} \\
&= \sum_{i=1}^{r} \sum_{T\in[T_i]} \frac{1}{\binom{2\kappa}{\kappa}} \Pr[S|p_T] \frac{\mathbb{1}(L_* \subseteq T^c)}{\binom{|T^c\cap M|}{\kappa/8}} \\
&= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \sum_{T\in[T_i]} \frac{\mathbb{1}(L_* \subseteq T^c)}{\binom{m_i'}{\kappa/8}} \\
&= \sum_{i=1}^{r} \frac{\Pr[S|p_{T_i}]}{\binom{2\kappa}{\kappa}} \frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}},
\end{aligned}$$

Here the only difference ends up being that we use m_i' instead of m_i since we have effectively replaced T with T^c. However, this replacement only takes place for q, the component of the probability that deals with S is identical for both Q and Q'.

Finally, based on these equations, it suffices to show that $\frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}} = \frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}}$. To do so,

since $m_i = |T_i \cap M|$ and $m_i' = |T_i^c \cap M|$, it follows that $m_i + m_i' = m$. Using this, we have that

$$\begin{aligned}\frac{\binom{m-\kappa/8}{m_i-\kappa/8}}{\binom{m_i}{\kappa/8}} &= \frac{(m-\kappa/8)!\,(\kappa/8)!(m_i-\kappa/8)!}{(m_i-\kappa/8)!(m-m_i)!m_i!} \\ &= \frac{(m-\kappa/8)!(\kappa/8)!}{m_i'!m_i!}.\end{aligned}$$

Applying the same manipulation to $\frac{\binom{m-\kappa/8}{m_i'-\kappa/8}}{\binom{m_i'}{\kappa/8}}$ completes the proof.

□

Step 4: finishing the overall proof.

Let κ be a sufficiently large integer. It suffices to show that there exists a probability distribution p with $\frac{1}{9\kappa} \leq p_\varepsilon \leq \frac{2}{3\kappa}$ such that $m_p(\varepsilon,\delta) > \kappa$. Assume towards a contradiction that no such p exists, meaning that $m_p(\varepsilon,\delta) \leq \kappa$ for all p satisfying the above.

Let $T \subset [2\kappa]$ satisfy $T = [2\kappa]$. By Lemma E.3.9, $\frac{1}{9\kappa} \leq (p_T)_\varepsilon \leq \frac{2}{3\kappa}$. It follows that with probability at least $1-\delta$ over $S \sim p_T^\kappa$ and $q \sim A_T(S)$ along with the randomness of B,

$$cr_q^{-\varepsilon} - \varepsilon \leq B(S,q) \leq cr_q^{\varepsilon} + \varepsilon,$$

with $cr_q^{-\varepsilon}, cr_q^{\varepsilon}$ denoting the appropriate data-copying rates for q with respect to p.

By Lemma E.3.13, if S covers T, then $cr_q^{-\varepsilon} \geq \frac{\lambda(1+\varepsilon)}{24} = \frac{13\frac{4}{3}}{24} > \frac{2}{3}$. By Lemma E.3.15, S covers T with probability at least $1 - 4\exp\left(-\frac{\kappa}{2048}\right)$. Substituting this, we have

$$\begin{aligned}1-\delta &\leq \mathbb{E}_{S\sim p_T^\kappa}\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B \mathbb{1}\left(B(S,q) \geq cr_q^{-\varepsilon} - \varepsilon\right) \\ &= \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ does not cover } T\right) + \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right) \\ &\leq 4\exp\left(-\frac{\kappa}{2048}\right) + \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_{q\sim A_T(S)}\mathbb{E}_B\mathbb{1}\left(B(S,q) > \frac{1}{3}\right),\end{aligned}$$

with the substitutions for $cr_q^{-\varepsilon} - \varepsilon$ utilizing that $\varepsilon = \frac{1}{3}$.

Applying this over the distribution, $\mathscr{F}$ (Definition E.3.16), which comprises of all

(p_T, A_T) with T chosen at uniform over all subsets of size κ, and then substituting the definition of Q (Definition E.3.17), we have

$$\begin{aligned}
1-\delta-4\exp\left(-\frac{\kappa}{2048}\right) &\leq \mathbb{E}_{(p_T,A_T)\sim\mathscr{F}}\mathbb{E}_{S\sim p_T^\kappa}\mathbb{E}_{q\sim A_T(S)}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_B\mathbb{1}\left(B(S,q)>\frac{1}{3}\right) \\
&= \mathbb{E}_{(S,q)\sim Q}\mathbb{1}\left((S,q) \text{ is nice}\right)\mathbb{E}_B\mathbb{1}\left(B(S,q)>\frac{1}{3}\right) \\
&= \frac{1}{1-\Pr_{(S,q)\sim Q}[(S,q) \text{ is not nice}]}\mathbb{E}_{(S,q)\sim Q_*}\mathbb{E}_B\mathbb{1}\left(B(S,q)>\frac{1}{3}\right) \\
&\leq \mathbb{E}_{(S,q)\sim Q_*}\mathbb{E}_B\mathbb{1}\left(B(S,q)>\frac{1}{3}\right),
\end{aligned} \tag{E.27}$$

where Q_* denotes the marginal distribution of Q over all nice (Definition E.3.18) pairs (S,q). Note that the manipulation above holds because of Lemma E.3.19, which implies that (S,q) is nice if and only if S covers T.

Next, we apply the same exact reasoning to the pair (p_T, A'_T). To this end, we have that with probability at least $1-\delta$ over $S\sim p_T^\kappa$, $q\sim A'_T(S)$, along with the randomness of B,

$$cr_q^{-\varepsilon}-\varepsilon \leq D(S,q) \leq cr_q^{\varepsilon}+\varepsilon.$$

By Lemma E.3.13, if S covers T, then $cr_q^\varepsilon = 0$. Applying the same argument as above using Lemma E.3.15, we have that

$$1-\delta-4\exp\left(-\frac{\kappa}{2048}\right) \leq \mathbb{E}_{S\sim p_T^\kappa}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_{q\sim A'_T(S)}\mathbb{E}_B\mathbb{1}\left(B(S,q)\leq\frac{1}{3}\right).$$

Applying this over the distribution $\mathscr{F}'$ (Definition E.3.16) and using a similar set of manipulations

as we did with $\mathscr{F}$ and Q, we have that

$$1-\delta-4\exp\left(-\frac{\kappa}{2048}\right) \leq \mathbb{E}_{(p_T,A'_T)\sim\mathscr{F}'}\mathbb{E}_{S\sim p_T^\kappa}\mathbb{E}_{q\sim A'_T(S)}\mathbb{1}\left(S \text{ covers } T\right)\mathbb{E}_B\mathbb{1}\left(B(S,q)\leq\frac{1}{3}\right)$$
$$\leq \mathbb{E}_{(S,q)\sim Q'_*}\mathbb{E}_B\mathbb{1}\left(B(S,q)\leq\frac{1}{3}\right), \tag{E.28}$$

where Q'_* denotes the marginal distribution of Q' over nice pairs (S,q).

Finally, by Lemma E.3.20, Q'_* and Q_* follow the exact same distribution. This means that summing equations E.27 and E.28, we can combine the summands *inside* the expectation giving us that

$$2-2\delta-8\exp\left(-\frac{\kappa}{2048}\right) \leq \mathbb{E}_{(S,q)\ sim Q_*}\mathbb{E}_B\left(\mathbb{1}\left(B(S,q)>\frac{1}{3}\right)+\mathbb{1}\left(B(S,q)\leq\frac{1}{3}\right)\right)=1.$$

This gives a contradiction as this equation is clearly false when κ is sufficiently large (as $\delta=\frac{1}{3}$).

□

Bibliography

[1] 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.

[2] 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.

[3] Casey Meehan, Kamalika Chaudhuri, and Sanjoy Dasgupta. A three sample hypothesis test for evaluating generative models. In Silvia Chiappa and Roberto Calandra, editors, *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pages 3546–3556. PMLR, 2020.

[4] 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.

[5] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, March 20 2014. URL http://arxiv.org/abs/1412.6572.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, 1996.

[13] Charles Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–645, 1977.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Hongge Chen, Huan Zhang, Duane S. Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. *CoRR*, abs/1902.10660, 2019.

[29] Geoffrey W. Gates. The reduced nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 18(3):431–433, 1972.

[30] 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.

[31] Peter E. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 14(3):515–516, 1968.

[32] 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.

[33] 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.

[34] Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Universal bayes consistency in metric spaces. *CoRR*, abs/1906.09855, 2019.

[35] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Information Theory*, 51(1):128–142, 2005.

[36] Yao-Yuan Yang, Cyrus Rashtchian, Ruslan Salakhutdinov, and Kamalika Chaudhuri. A closer look at robustness vs. accuracy. In *Neural Information Processing Systems (NeuRIPS)*, 2020.

[37] 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.

[38] 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.

[39] 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.

[40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Mach. Learn.*, 20(3): 273–297, 1995.

[41] Kristin P. Bennett and Erin J. Bredensteiner. 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.

[42] Yoav Freund and Robert E. Schapire. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3):277–296, 1999.

[43] 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.

[44] 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.

[45] 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.

[46] 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.

[47] Sadia Chowdhury and Ruth Urner. On the (un-)avoidability of adversarial examples. *CoRR*, abs/2106.13326, 2021. URL https://arxiv.org/abs/2106.13326.

[48] 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.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] Vladimir N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1998.

[59] Robi Bhattacharjee and Kamalika Chaudhuri. Consistent non-parametric methods for

maximizing robustness. In Marc'Aurelio Ranzato, Alina Beygelzimer, Yann N. Dauphin, Percy Liang, and Jennifer Wortman Vaughan, editors, *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pages 9036–9048, 2021.

[60] Hassan Ashtiani, Vinayak Pathak, and Ruth Urner. Adversarially robust learning with tolerance. *CoRR*, abs/2203.00849, 2022. doi: 10.48550/arXiv.2203.00849. URL https://doi.org/10.48550/arXiv.2203.00849.

[61] Omar Montasser, Steve Hanneke, and Nathan Srebro. Transductive robust learning guarantees. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *International Conference on Artificial Intelligence and Statistics, AISTATS 2022, 28-30 March 2022, Virtual Event*, volume 151 of *Proceedings of Machine Learning Research*, pages 11461–11471. PMLR, 2022.

[62] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014. ISBN 978-1-10-705713-5.

[63] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974.

[64] David Lopez-Paz and Maxime Oquab. Revisiting classifier two-sample tests. *arXiv preprint arXiv:1610.06545*, 2016.

[65] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Q. Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *CoRR*, abs/1806.07755, 2018. URL http://arxiv.org/abs/1806.07755.

[66] Gavin Brown, Mark Bun, Vitaly Feldman, Adam Smith, and Kunal Talwar. When is memorization of irrelevant training data necessary for high-accuracy learning? In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, jun 2021. doi: 10.1145/3406325.3451131. URL https://doi.org/10.1145%2F3406325.3451131.

[67] Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy, editors, *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, pages 954–959. ACM, 2020. doi: 10.1145/3357713.3384290. URL https://doi.org/10.1145/3357713.3384290.

[68] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 International Joint Conference on Neural Networks, IJCNN 2020, Glasgow, United Kingdom, July 19-24, 2020*, pages 1–10. IEEE, 2020. doi: 10.1109/IJCNN48605.2020.9207181. URL https://doi.org/10.1109/IJCNN48605.2020.9207181.

[69] Yasin Yazici, Chuan-Sheng Foo, Stefan Winkler, Kim-Hui Yap, and Vijay Chandrasekhar.

Empirical analysis of overfitting and mode drop in gan training. In *IEEE International Conference on Image Processing, ICIP 2020, Abu Dhabi, United Arab Emirates, October 25-28, 2020*, pages 1651–1655. IEEE, 2020. doi: 10.1109/ICIP40778.2020.9191083. URL https://doi.org/10.1109/ICIP40778.2020.9191083.

[70] Mehdi S. M. Sajjadi, Olivier Bachem, Mario Lucic, Olivier Bousquet, and Sylvain Gelly. Assessing generative models via precision and recall. 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, December 3-8, 2018, Montréal, Canada*, pages 5234–5243, 2018.

[71] Tim Salimans, Ian J. Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *CoRR*, abs/1606.03498, 2016. URL http://arxiv.org/abs/1606.03498.

[72] Eitan Richardson and Yair Weiss. On gans and gmms. 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, December 3-8, 2018, Montréal, Canada*, pages 5852–5863, 2018.

[73] Satrajit Chatterjee. Learning and memorization. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 755–763. PMLR, 10–15 Jul 2018.

[74] Gerrit van den Burg and Chris Williams. On memorization in probabilistic deep generative models. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 27916–27928. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/eae15aabaa768ae4a5993a8a4f4fa6e4-Paper.pdf.

[75] Ching-Yuan Bai, Hsuan-Tien Lin, Colin Raffel, and Wendy Chi-wen Kan. On training sample memorization: Lessons from benchmarking generative modeling with a large-scale competition. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao, editors, *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pages 2534–2542. ACM, 2021. doi: 10.1145/3447548.3467198. URL https://doi.org/10.1145/3447548.3467198.

[76] Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramèr, and Chiyuan Zhang. Quantifying memorization across neural language models. *CoRR*, abs/2202.07646, 2022. URL https://arxiv.org/abs/2202.07646.

[77] Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret

sharer: Evaluating and testing unintended memorization in neural networks. In Nadia Heninger and Patrick Traynor, editors, *28th USENIX Security Symposium, USENIX Security 2019, Santa Clara, CA, USA, August 14-16, 2019*, pages 267–284. USENIX Association, 2019.

[78] Cynthia Dwork. Differential privacy. In Michele Bugliesi, Bart Preneel, Vladimiro Sassone, and Ingo Wegener, editors, *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006, Venice, Italy, July 10-14, 2006, Proceedings, Part II*, volume 4052 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2006.

[79] Adam Block, Zeyu Jia, Yury Polyanskiy, and Alexander Rakhlin. Intrinsic dimension estimation using wasserstein distances. *Journal of machine learning research*, 1533-7928, 2022.

[80] Robert B. Ash. *Information theory*. Dover Publications, 1990.

[81] Sanjoy Dasgupta, Daniel J. Hsu, and Claire Monteleoni. A general agnostic active learning algorithm. In John C. Platt, Daphne Koller, Yoram Singer, and Sam T. Roweis, editors, *Advances in Neural Information Processing Systems 20, Proceedings of the Twenty-First Annual Conference on Neural Information Processing Systems, Vancouver, British Columbia, Canada, December 3-6, 2007*, pages 353–360. Curran Associates, Inc., 2007.

[82] Norbert Sauer. On the density of families of sets. *Journal of Combinatorial Theory, Series A*, 13(1):145–147, 1972.

[83] Saharon Shelah. A combinatorial problem; stability and order for models and theories in infinitary languages. *Pacific Journal of Mathematics*, 41(1):247–261, 1972.