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

Above all, I thank my advisor, Kamalika, for her mentorship over the last five years. Kamalika will do anything in her power to help her students find what success means to them, and build towards it. She puts her students' futures above all else. Part of that is mentoring her students to become thoughtful researchers. In hindsight, I see that we could have pursued much lower risk projects that made incremental gains on the hottest research directions. Instead, Kamalika guided me towards unusual and challenging problems that required me to examine my fundamentals and reconsider broad questions about the aims of data privacy. In doing so, I not only understand what solutions are effective in data privacy, but why we as a research community use them and — in some cases — why we have ignored them.

I cannot overemphasize the gratitude I have to my collaborators as well. First, to my collaborators at FAIR. I thank Chuan Guo for showing me how to navigate incredibly challenging and open-ended ML problems. His Socratic approach to mentoring helped me explore my own instincts and take ownership of the project without letting me slip down the wrong path. Florian Bordes taught me the ins and outs of contemporary vision modeling. Pascal Vincent's insights on how to design deep learning experiments were truly formative; I sincerely appreciate his effort and time. I owe an enormous debt of gratitude to Ashwin Machanavajjhala at Tumult for mentoring me on how how to reason about different privacy definitions and guarantees in applied settings.

Finally, to all of my friends. To the lab: I could not have landed in a better group. To be surrounded by such wonderful and bright companions is a win — for them to be your colleagues as well is a gift. To all my dear pre-PhD friends (you know who you are, and you really don't have read this dissertation): you are the loves of my life!

Chapter 1, in full, has been submitted for publication of the material as it may appear in Neural Information Processing Systems, 2023, Casey Meehan, Florian Bordes, Pascal Vincent, Kamalika Chaudhuri, Chuan Guo. *Do SSL models have Déjà Vu? A Case of Unintended Memorization in Self-Supervised Learning*. The dissertation author shares equal contribution with Florian Bordes.

Chapter 2, in full, is a reprint of the material as it appears in International Conference on Artificial Intelligence and Statistics, 2020. Casey Meehan, Sanjoy Dasgupta, Kamalika Chaudhuri. *A Non-parametric Test to Detect Data-Copying in Generative Models*. The dissertation author is the primary investigator and author of this paper.

Chapter 3, in full, is a reprint of the material as it appears in Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022. Casey Meehan, Khalil Mrini, Kamalika Chaudhuri. *Sentence-level Privacy for Document Embeddings*. The dissertation author is the primary investigator and author of this paper.

Chapter 4, in full, is a reprint of the material as it appears in International Conference on Learning Representations. 2022. Casey Meehan, Amrita Roy-Chowdhury, Kamalika Chaudhuri, Somesh Jha. *Privacy Implications of Shuffling*. The dissertation author is the primary investigator and author of this paper.

Chapter 5, in full, is a reprint of the material as it appears in International Conference on Artificial Intelligence and Statistics, 2021. Casey Meehan, Kamalika Chaudhuri. *Location Trace Privacy Under Conditional Priors*. The dissertation author is the primary investigator and author of this paper.

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.

Introduction

In machine learning (ML) we learn broad trends and patterns from vast sums of data. What type of data is collected and how it is used introduces different kinds of privacy risks. Medical data allows us to link characteristics like genetic markers to adverse health conditions like cancer. However it suffers from unique correlation risks, since one's information can reliably be correlated from their familys'. Document embeddings—useful vector representations of text documents—are tremendously useful for inferring general document characteristics like topic, but can expose detailed personal information at the sentence level. Generative vision models can sample novel images, but also tend to copy and expose their training images in clever and inexact ways.

The privacy risks and utility requirements of each of these settings and applications warrant different approaches to privacy-preserving ML. This dissertation proposes a variety of solutions to situations like those above. In doing so, I hope to illuminate the advantage of taking an application-specific approach to both measuring privacy risks and engineering private algorithms. The following five chapters are roughly organized into two parts: the first two chapters cover *empirical* privacy methods, and the remaining three cover *formal* privacy methods. The former includes statistical tests to quantify privacy risks of large ML models. The latter proposes provably private algorithms to satisfy different privacy definitions chosen for different ML tasks.

Empirical methods. The first two chapters explore empirically measuring privacy risks in the domain of vision modeling. In both chapters, we analyze to what extent large vision models *memorize* their training images, and thereby risk exposing them. In contrast with the following three chapters, we are not proposing a provably privacy-preserving algorithm. Instead we are designing methodical empirical tests to quantify memorization.

Provably private algorithms. The final three chapters propose algorithms that allow us to share our data in a provably private way. We explore privacy preserving algorithms in the text, location, and interpersonal-correlated domains (*e.g.* social networks or genetically linked medical data). For each of these, we study different ML tasks and privacy risks to motivate different privacy definitions and provably private algorithms.

Taken together, this document offers a mindset towards practicable privacy methods. By directly considering the data domain and the task at hand, it is possible to efficiently measure privacy risks and propose provably private algorithms while still completing the learning task at hand .

Chapter 1

When are Non-Parametric Methods Robust?

1.1 Introduction

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

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

In this work, we consider this problem in the context of general non-parametric classifiers. Contrary to parametrics, non-parametric methods are a form of local classifiers, and include a large number of pattern recognition methods such as nearest neighbors, decision trees, random forests and kernel classifiers. There is a richly developed statistical theory of non-parametric methods [9], which focuses on accuracy, and provides very general conditions under which these methods converge to the Bayes optimal with growing number of samples. We, in contrast, analyze robustness properties of these methods, and ask instead when they converge to the classifier with the highest astuteness at a desired radius r. Recall that the astuteness of a classifier at radius r is the fraction of points from the distribution on which it is accurate and has the same prediction up to a distance r [5, 4].

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

This raises the natural question – when do non-parametric methods produce astute classifiers? We next provide conditions under which a non-parametric method converges to the most astute classifier in the large sample limit under well-separated data. Our conditions are analogous to the classical conditions for convergence to the Bayes optimal [9, 10], but a little stronger. We show that nearest neighbors and kernel classifiers whose kernel functions decay fast enough, satisfy these conditions, and hence converge to astute classifiers in the large sample limit. In 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, [11] proposes a method called Adversarial Pruning that preprocesses the training data by retaining the maximal set of points such that different classes are distance $\geq 2r$ apart, and then trains a non-parametric method on the pruned data. We next prove that if a non-parametric method has certain properties, then the classifier produced by Adversarial Pruning followed by the method does 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 [11] by providing a performance guarantee.

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

1.1.1 Related Work

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

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

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

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

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

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

1.2 Preliminaries

1.2.1 Setting

We consider binary classification where instances are drawn from a totally bounded metric space $\mathscr{X}$ 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. *(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 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* [5, 4].

Definition 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 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 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 [5], is that the classifier should converge towards being astute where the Bayes Optimal classifier is astute. However, the Bayes Optimal classifier is not necessarily the most astute classifier and may even have poor astuteness. To see this, consider the following example.

Example 1

Consider $\mathscr{D}$ 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 [11], which is the classifier with maximum astuteness.

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

$$g_r^* = \operatorname*{arg\,max}_{f\in\{\pm 1\}^{\mathscr{X}}} A_r(f,\mathscr{D}).$$

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 7. *(r-consistent) Let M be a classification algorithm over $\mathscr{X}\times\{\pm 1\}$. 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 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 9. *[9] A* ***weight function*** *W is a non-parametric classifier with the following properties.*

1. *Given input $S = \{(x_1, y_1), (x_2, y_2,), \dots, (x_n, y_n)\} \sim \mathscr{D}^n$, W constructs functions $w_1^S, \dots, w_n^S : \mathscr{X} \to [0,1]$ 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 $x_1, x_2, \dots x_n$ but must be independent of $y_1, y_2, \dots, y_n$.*
2. *W has output W_S defined as*

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

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

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

Definition 10. *A **histogram classifier**, H, is a non-parametric classification algorithm over $\mathbb{R}^d \times \{\pm 1\}$ that works as follows. For a distribution $\mathscr{D}$ 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 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.

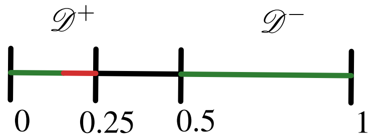

Figure 1.1. H_S is astute in the green region, but not robust in the red region.

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

Theorem 12. *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 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathscr{D}^n$. 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 12 to Stone's theorem [10], which gives sufficient conditions for a weight function to be consistent (i.e. converge in accuracy towards the Bayes optimal). For convenience, we include a statement of Stone's theorem.

Theorem 13. *[10] Let W be weight function over $\mathscr{X} \times \{\pm 1\}$. 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 $S = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\} \sim \mathscr{D}^n$. 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 12 and Stone's theorem.

1. Conditions 1. and 3. of Stone's theorem are no longer necessary. This is because r-separated distributions are well-separated and thus have simpler conditions for consistency. In fact, a slight modification of the arguments of [10] shows that for r-separated distributions, condition 2. alone is sufficient for consistency.

2. Condition 2. is strengthened. Instead of requiring the weight of x_i's outside of a given radius to go to 0 for $X \sim \mathscr{D}$, we require the same to *uniformly* hold over a ball centered at X.

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

Corollary 14. *Let $\mathscr{D}$ 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 15. *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 15 necessary? The following example illustrates that a kernel classifier with any arbitrary K is not necessarily r-consistent. This indicates that some sort of condition needs to be imposed on K to ensure r-consistency; finding a tight necessary condition however is left for future work.

Example 3

Let $\mathscr{X} = [-1, 1]$ 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 13 (the proof is analogous to the case for a Gaussian Classifier), and is therefore consistent. However, W does not learn a robust classifier on $\mathscr{D}$ 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 $1 \leq i \leq n$. Since $x_i = 1$ with probability 0.9, it follows that with high probability x will be classified as 1 which means that f, the output of W, is not robust at $x = -1$. Thus f has astuteness at most 0.9 which means that W is *not* r-consistent for $r = 0.3$.

1.4 General Distributions

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

1.4.1 The r-Optimal Classifier and Adversarial Pruning

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

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

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

Definition 16. *[11] 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 $d(S^*_j,x) \leq r$.*

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

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

Definition 17. *A set $S_r \subset \mathscr{X} \times \{\pm 1\}$ 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 [11] show good empirical performance of such algorithms, no formal guarantees are provided. We next formally characterize when adversarial pruning followed by a non-parametric method results in a classifier that is provably r-consistent.

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

Algorithm 1: RobustNonPar

```
Input: $S \sim \mathscr{D}^n$, weight function $W$, robustness radius $r$;
$S_r \leftarrow AdvPrun(S, r)$;
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 18. *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 18 and Theorem 12.

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

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

Remark:

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

Corollary 20. *Let W be a kernel classifier over $\mathscr{X}\times\{\pm1\}$ 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. $\lim_{n \to \infty} h_n = 0$.

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

Concluding Remarks

The above chapters provide a diverse set of examples of how privacy risks can be measured or mitigated in different scenarios. While highly different from each other, these examples all highlight the following three guiding principles for data privacy.

No privacy definition is a 'gold standard'

Differential privacy (DP) is often touted as the 'gold standard' of data privacy. The cases studied in Chapters 3-5 challenge this by proposing entirely different privacy definitions for entirely different settings and risks. Chapter 3 proposes sentence privacy, which is DP-like but uses a different neighboring notion. Chapter 4, on the other hand, proposes a shuffling-based privacy definition that is almost orthogonal to DP. That is because the correlation adversary considered in that setting cannot be thwarted by DP alone. However, the broad population trends that we wish to learn are still accessible under our semi-random shuffling approach. Similarly, Chapter 5 analyzes the threat of correlation adversaries in the domain of location traces. Here, we see that a DP-based definition has to add *more noise* in order to thwart these adversaries.

Taken together, we see that sometimes DP definitions are effective, and other times they require one to choose between meaningful privacy and utility. If one 'gold standard' definition were effective in all of these settings, we would not need to propose so many contrasting privacy definitions and methods.

No Free Lunch

The goal of data privacy is to allow the release of high-level information (*e.g.* data distribution) while obscuring low-level information (*e.g.* individuals' data features). It is natural to wonder whether it is possible to design a privacy definition under which we can release highly level information and defend against *any* adversary. The answer is unequivocally, *no*. This fact is known as the No-Free-Lunch theorem, made precise in [**?**]. The Theorem shows that releasing any information about a dataset that is useful to one person can be leveraged by an adversary to learn fine-grained information. The No-Free-Lunch theorem is instructive, because it shifts our attention from the question of whether we can provide air-tight privacy (impossible) to whether the adversaries our definition allows are *realistic* in our setting.

The No-Free-Lunch principle is fundamental to the approaches of Chapters 4 and 5 in particular. Here, we propose novel privacy definitions that are adversary-focused. Note that in both of these papers, we consider limited classes of adversaries. As stated above, it is impossible to block the inferences of *all adversaries* while still sharing useful information derived from the sensitive data. By practically evaluating what prior knowledge an adversary might have, like a correlation prior, we can formalize a privacy definition that gives strong guarantees in realistic settings.

Perfect is the enemy of good

Chapters 1 and 2 offer no formal privacy definition or provably private mechanism. Instead, they offer statistical tests to empirically evaluate a model's memorization of its training data, and thereby risk of exposing that data. In both chapters, we examine how model selection can effect the degree of memorization as detected by our tests. While our proposed test statistics do not confer any formal privacy guarantees, they guide practitioners towards models that memorize less. In many cases, our tests showed that it is possible to find models which have significantly less memorization at little to no cost in utility.

While formal privacy definitions are a valuable goal they make up only a small part of an ML practitioners privacy toolkit. To preserve privacy, we as researchers ought to put equal effort into methodical empirical privacy tests as we do formally private algorithms. These tests tend to be far more accessible to practitioners and allow them to significantly improve model privacy. Although empirical privacy tests are imperfect, the practical benefits to be gained by proposing them are undoubtedly a positive good. Do not let perfect privacy be the enemy of good privacy.

Bibliography

[1] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[2] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, March 20 2014. URL http://arxiv.org/abs/1412.6572.

[3] Daniel Lowd and Christopher Meek. Adversarial learning. In Robert Grossman, Roberto J. Bayardo, and Kristin P. Bennett, editors, *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, Illinois, USA, August 21-24, 2005*, pages 641–647. ACM, 2005. ISBN 1-59593-135-X.

[4] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *Advances in Neural Information Processing Systems 31*, pages 5014–5026. Curran Associates, Inc., 2018.

[5] Yizhen Wang, Somesh Jha, and Kamalika Chaudhuri. Analyzing the robustness of nearest neighbors to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, pages 5120–5129, 2018.

[6] Omar Montasser, Steve Hanneke, and Nathan Srebro. VC classes are adversarially robustly learnable, but only improperly. In *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, pages 2512–2530, 2019.

[7] Hadi Salman, Greg Yang, Jerry Li, Pengchuan Zhang, Huan Zhang, Ilya P. Razenshteyn, and Sébastien Bubeck. Provably robust deep learning via adversarially trained smoothed classifiers. *CoRR*, abs/1906.04584, 2019. URL http://arxiv.org/abs/1906.04584.

[8] Dmitrii Avdiukhin, Slobodan Mitrovic, Grigory Yaroslavtsev, and Samson Zhou. Adversarially robust submodular maximization under knapsack constraints. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019.*, pages 148–156, 2019. doi:

10.1145/3292500.3330911.

[9] Luc Devroye, László Györfi, and Gábor Lugosi. *A Probabilistic Theory of Pattern Recognition*, volume 31 of *Stochastic Modelling and Applied Probability*. Springer, 1996.

[10] Charles Stone. Consistent nonparametric regression. *The Annals of Statistics*, 5(4):595–645, 1977.

[11] Yao-Yuan Yang, Cyrus Rashtchian, Yizhen Wang, and Kamalika Chaudhuri. Adversarial examples for non-parametric methods: Attacks, defenses and large sample limits. *CoRR*, abs/1906.03310, 2019. URL http://arxiv.org/abs/1906.03310.

[12] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57, 2017.

[13] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.

[14] Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *ASIACCS*, 2017.

[15] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *IEEE European Symposium on Security and Privacy, EuroS&P 2016, Saarbrücken, Germany, March 21-24, 2016*, pages 372–387, 2016.

[16] Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2266–2276. Curran Associates, Inc., 2017.

[17] Guy Katz, Clark W. Barrett, David L. Dill, Kyle Julian, and Mykel J. Kochenderfer. Towards proving the adversarial robustness of deep neural networks. In *Proceedings First Workshop on Formal Verification of Autonomous Vehicles, FVAV@iFM 2017, Turin, Italy, 19th September 2017.*, pages 19–26, 2017.

[18] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[19] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016*, pages 582–597, 2016.

[20] Aditi Raghunathan, Jacob Steinhardt, and Percy Liang. Certified defenses against adversarial examples. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[21] Aman Sinha, Hongseok Namkoong, and John C. Duchi. Certifying some distributional robustness with principled adversarial training. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018.

[22] Laurent Amsaleg, James Bailey, Dominique Barbe, Sarah M. Erfani, Michael E. Houle, Vinh Nguyen, and Milos Radovanovic. The vulnerability of learning to adversarial perturbation increases with intrinsic dimensionality. In *2017 IEEE Workshop on Information Forensics and Security, WIFS 2017, Rennes, France, December 4-7, 2017*, pages 1–6, 2017.

[23] Chawin Sitawarin and David A. Wagner. On the robustness of deep k-nearest neighbors. In *2019 IEEE Security and Privacy Workshops, SP Workshops 2019, San Francisco, CA, USA, May 19-23, 2019*, pages 1–7, 2019.

[24] Maksym Andriushchenko and Matthias Hein. Provably robust boosted decision stumps and trees against adversarial attacks. In H. Wallach, H. Larochelle, A. Beygelzimer, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12997–13008. Curran Associates, Inc., 2019.

[25] Alex Kantchelian, J. D. Tygar, and Anthony D. Joseph. Evasion and hardening of tree ensemble classifiers. *CoRR*, abs/1509.07892, 2015. URL http://arxiv.org/abs/1509.07892.

[26] Hongge Chen, Huan Zhang, Duane S. Boning, and Cho-Jui Hsieh. Robust decision trees against adversarial examples. *CoRR*, abs/1902.10660, 2019.

[27] Geoffrey W. Gates. The reduced nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 18(3):431–433, 1972.

[28] Lee-Ad Gottlieb, Aryeh Kontorovich, and Pinhas Nisnevitch. Near-optimal sample compression for nearest neighbors. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 370–378, 2014.

[29] Peter E. Hart. The condensed nearest neighbor rule (corresp.). *IEEE Trans. Information Theory*, 14(3):515–516, 1968.

[30] Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Nearest-neighbor sample compression: Efficiency, consistency, infinite dimensions. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 1573–1583, 2017.

[31] Aryeh Kontorovich and Roi Weiss. A bayes consistent 1-nn classifier. In *Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2015, San Diego, California, USA, May 9-12, 2015*, 2015.

[32] Steve Hanneke, Aryeh Kontorovich, Sivan Sabato, and Roi Weiss. Universal bayes consistency in metric spaces. *CoRR*, abs/1906.09855, 2019.

[33] Ingo Steinwart. Consistency of support vector machines and other regularized kernel classifiers. *IEEE Trans. Information Theory*, 51(1):128–142, 2005.