Sample Complexity of Robust Linear Classification on Separated Data

Robi Bhattacharjee [1] **Somesh Jha** [2] **Kamalika Chaudhuri** [1]

Abstract

We consider the sample complexity of learning with adversarial robustness. Most prior theoretical results for this problem have considered a setting where different classes in the data are close together or overlapping. We consider, in contrast, the well-separated case where there exists a classifier with perfect accuracy and robustness, and show that the sample complexity narrates an entirely different story. Specifically, for linear classifiers, we show a large class of well-separated distributions where the expected robust loss of any algorithm is at least $\Omega(\frac{d}{n})$, whereas the max margin algorithm has expected standard loss $O(\frac{1}{n})$. This shows a gap in the standard and robust losses that cannot be obtained via prior techniques. Additionally, we present an algorithm that, given an instance where the robustness radius is much smaller than the gap between the classes, gives a solution with expected robust loss is $O(\frac{1}{n})$. This shows that for very well-separated data, convergence rates of $O(\frac{1}{n})$ are achievable, which is not the case otherwise. Our results apply to robustness measured in any ℓ_p norm with $p > 1$ (including $p = \infty$).

1. Introduction

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

In this work, we ask: how many samples are needed to learn a classifier with low robust loss when $\mathcal{H}$ is the class of linear classifiers, and d is an ℓ_p-metric? Prior work has provided both upper (Yin et al., 2019; Dan et al., 2020) as well as lower bounds (Schmidt et al., 2018; Dan et al., 2020) 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 (Yang et al., 2020), 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 (Cullina et al., 2018) 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

*Equal contribution [1]University of California, San Diego [2]University of Wisconsin-Madison. Correspondence to: Robi Bhattacharjee <rcbhatta@eng.ucsd.edu>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

seeing a certain number of samples close to the margin.

For upper bounds, prior work (Yin et al., 2019) 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 $\Omega(d/\epsilon)$ 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

1.1. Related Work

There is a large body of work (Carlini & Wagner, 2017; Liu et al., 2017; Papernot et al., 2017; 2016a; Szegedy et al., 2014; Hein & Andriushchenko, 2017; Katz et al., 2017; Papernot et al., 2016b; Raghunathan et al., 2018; Sinha et al., 2018) empirically studying adversarial examples primarily in the context of neural networks. Several works (Schmidt et al., 2018; Raghunathan et al., 2020; Tsipras et al., 2019) 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 (Khim & Loh, 2018; Attias et al., 2019; Montasser et al., 2019; Yin et al., 2019; Ashtiani et al., 2020) have focused on finding distribution agnostic bounds of the sample complexity for robust classification. In (Montasser et al., 2019), 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.

(Diakonikolas et al., 2020) considers learning linear classifiers robustly, but is primarily focused on computational complexity as opposed to sample complexity.

In (Yin et al., 2019), 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 (Schmidt et al., 2018), 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 (Bhagoji et al., 2019), (Dobriban et al., 2020) and (Dan et al., 2020). (Bhagoji et al., 2019) 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. (Dobriban et al., 2020) applies expands upon (Schmidt et al., 2018) by consider mixtures of three Gaussians in both the ℓ_2 and ℓ_∞ norms. Finally, (Dan et al., 2020) fully generalizes the results of (Schmidt et al., 2018) 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 $p \in [1, \infty]$). (Schmidt et al., 2018) and (Dan et al., 2020) bear the most relevance with our work, and we consequently carefully compare our results in section 3.1.

Another approach for lower and upper bounds on sample complexities for linear classifiers can be found in (Cullina et al., 2018), 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, (Cullina et al., 2018) 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. (Wang et al., 2018) considers the nearest neighbors algorithm, and shows how to adapt it for converging towards a robust classifier. In (Yang et al., 2019), 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. (Bhattacharjee & Chaudhuri, 2020) 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.

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, $diam_p(S) = \sup_{x,y \in S} ||x - y||_p$.

2.1. Standard and Robust Loss

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

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

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

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

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

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

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 4. *Let A be a learning algorithm and let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. For any $S \sim \mathcal{D}^n$, we let A_S denote the classifier learned by A from training data S. Then the **expected standard loss** of A with respect to $\mathcal{D}$, denoted $EL^n(A, \mathcal{D})$ where n is the number of training samples, is defined as*

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

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

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

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

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

Definition 5. *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ and A be a learning algorithm. Then the **standard sample complexity** of A with respect to $\mathcal{D}$, denoted $m^\epsilon(A, \mathcal{D})$, is the*

minimum number of training samples needed such that A has expected standard loss at most ϵ. Formally,

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

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

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

2.3. Linear classifiers

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

Definition 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 7. *A distribution $\mathcal{D}$ 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, $P_{(x,y)\sim\mathcal{D}}[x \in S^y] = 1$.

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

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

Definition 8. *Let $\mathcal{D}$ be a linearly separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let S^+ and S^- be as above. Then $\mathcal{D}$ 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, $\mathcal{L}(f_{w,b}, \mathcal{D}) = 0$.

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 $\gamma(\mathcal{D})$ denote the margin of $\mathcal{D}$.

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

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

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 10. (Chapter 10 in (Vapnik, 1998)) *Let M denote the hard margin SVM algorithm. If $\mathcal{D}$ is a distribution with aspect ratio $\rho = \rho(\mathcal{D})$, then for any $n > 0$ we have $\mathbb{E}_{S\sim\mathcal{D}^n}\mathcal{L}(M_S, \mathcal{D}) \leq O(\frac{\rho^2}{n})$, 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 11. *Let M denote the hard margin SVM algorithm. If $\mathcal{D}$ is a distribution with aspect ratio $\rho = \rho(\mathcal{D})$, then for any $\epsilon > 0$ we have $m^{\epsilon}(M_S, \mathcal{D}) \leq O(\frac{\rho^2}{\epsilon})$, where M_S denotes the classifier learned by M from training data S.*

Theorem 10 and Corollary 11 will serve as a benchmark for comparison with the robust sample complexity.

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

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 10.

3. Lower Bounds

In this section, we consider r-separated distributions whose aspect ratio is constant. By Theorem 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 13. *For any ρ, r, the set $\mathcal{F}_{r,\rho}$ is defined as the set of all distributions $\mathcal{D}$ over $\mathbb{R}^d \times \{\pm 1\}$ such that $\mathcal{D}$ is r-separated and has aspect ratio at most ρ.*

We now state our main result.

Theorem 14. *Let $r > 0$ and $\rho > 20$. Then the following hold.*

1. *For every learning algorithm A, and any $n > 0$, there exists $\mathcal{D} \in \mathcal{F}_{r,\rho}$ such that the expected robust loss when A is trained on a sample of size n from $\mathcal{D}$ is at least $\Omega(\frac{d}{n})$. Formally, there exists a constant $c > 0$ such that $\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})] \geq \frac{cd}{n}$.*
2. *In contrast, by Theorem 10, for any $\mathcal{D} \in \mathcal{F}_{r,D}$, the max margin algorithm has expected standard loss $O(\frac{\rho^2}{n})$, when trained on a sample of size n from $\mathcal{D}$. Formally, there exists a constant $c' > 0$ such that $\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}(A_S, \mathcal{D})] \leq \frac{c'\rho^2}{n}$.*

The condition $\rho > 20$ is required to rule out degenerate cases. This is because for small values of ρ, the ℓ_2 diameter of $\mathcal{D}$ is not much larger than the ℓ_2 margin of $\mathcal{D}$. This forces $\mathcal{D}$ 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 $\mathcal{D} \in \mathcal{F}_{r,\rho}$ for which the lower bound holds is given in terms of the norm p. More generally, the family $\mathcal{F}_{r,\rho}$ 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 (Cullina et al., 2018) because it specifically holds for $\mathcal{F}_{r,\rho}$, a linearly r-separated family of distributions with constant aspect ratio. Thus, while (Cullina et al., 2018) has shown the existence of distributions satisfying the first condition of Theorem 14, our result is the first to exhibit a distribution satisfying both conditions.

Finally, we note that Theorem 14 can also be expressed in terms of sample complexities. We include this in the following corollary.

Corollary 15. *Let $r > 0$ and $\rho > 20$. Then the following hold.*

1. *For every learning algorithm A, and any $\epsilon > 0$, there exists $\mathcal{D} \in \mathcal{F}_{r,\rho}$ such that the robust sample complexity of A with respect to $\mathcal{D}$ is at least $\Omega(\frac{d}{\epsilon})$. Formally, there exists a constant $c > 0$ such that $m_r^\epsilon(A, \mathcal{D}) \geq \frac{cd}{\epsilon}$.*
2. *In contrast, by Theorem 10, for any $\mathcal{D} \in \mathcal{F}_{r,D}$, the max margin algorithm has standard sample complexity $O(\frac{\rho^2}{\epsilon})$. Formally, there exists a constant $c' > 0$ such that $m^\epsilon(A, \mathcal{D}) \leq \frac{c'\rho^2}{\epsilon}$.*

3.1. Comparison with (Dan et al., 2020) and (Schmidt et al., 2018)

The first work to provide a robust sample complexity lower bound that applied to linear classifiers is (Schmidt et al., 2018); 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 (Dan et al., 2020), who also established more general lower bounds for any ℓ_p norm. Since (Dan et al., 2020) is a strict generalization of (Schmidt et al., 2018), we next explain how our lower bounds differ from (Dan et al., 2020), and why their techniques do not lead to our results. We begin by summarizing their results.

Summary of (Dan et al., 2020) (Dan et al., 2020) considers data distributions $\mathcal{D}$ that are parametrized by $\mu \in \mathbb{R}^d$ and $\Sigma \in \mathbb{R}^{d\times d}$, $\Sigma \succcurlyeq 0$. $\mathcal{D}_{\mu,\Sigma}$ is the mixture of two Gaussians, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(-\mu, \Sigma)$, with equal mass, where instances drawn from $\mathcal{N}(\mu, \Sigma)$ are labeled as $+$, and instances drawn from $\mathcal{N}(-\mu, \Sigma)$ 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 $\mathcal{D}_{\mu,\Sigma}$, let L_{rob} denote the optimal robust loss of any classifier on $\mathcal{D}_{\mu,\Sigma}$, and let L_{std} denote the optimal standard loss. Then the bounds shown in (Dan et al., 2020) can restated as follows (a detailed derivation from (Dan et al., 2020) appears in Appendix A).

Theorem 16. *(Dan et al., 2020)*

1. *For any learning algorithm A and any $n > 0$, there exists some mixture of Gaussians, $\mathcal{D}_{\mu,\Sigma}$ 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 $\mathcal{D}$.*
2. *For any distribution $\mathcal{D}_{\mu,\Sigma}$, 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 16, we see that although (Dan et al., 2020) 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 A.

3.2. Intuition behind Theorem 14

The proof idea for Theorem 14 can be summarized with a simple example (Figure 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 $S^+ \cup S^-$.

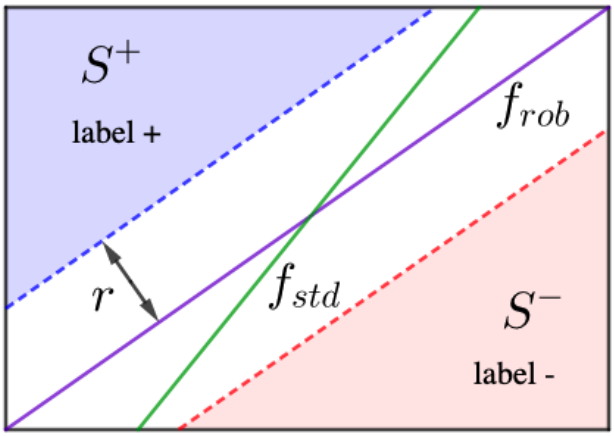

Figure 1. An example of a linearly r-separated distribution, with positively and negatively labeled examples in S^+ and S^- respectively. The optimally robust classifier, f_{rob} is shown in purple, while the (not necessarily unique) optimally accurate classifier, f_{std}, is shown in green.

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

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

and let

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

$\mathcal{H}_{\mathcal{D},\alpha}$ can be thought of as the set of accurate classifiers while $\mathcal{H}^r_{\mathcal{D},\alpha}$ can be thought of as the set of astute classifiers. By their definitions, it is clear that $\mathcal{H}^r_{\mathcal{D},\alpha} \subseteq \mathcal{H}_{\mathcal{D},\alpha}$. However, in the case when $\mathcal{H}$ is the set of linear classifiers, we see that for small α, $\mathcal{H}^r_{\mathcal{D},\alpha}$ is a much "smaller" set than $\mathcal{H}_{\mathcal{D},\alpha}$. By exploiting the geometric structure inherent to $\mathcal{H}$, we can much more efficiently search for some $h \in \mathcal{H}_{\mathcal{D},\alpha}$ than we can in $\mathcal{H}^r_{\mathcal{D},\alpha}$. 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 $\mathcal{H}^r_{\mathcal{D},\alpha}$ that aren't required for $\mathcal{H}_{\mathcal{D},\alpha}$.

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

Algorithm 1 Adversarial-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathcal{D}^n$,
$w \leftarrow 0$
for $i = 1 \ldots n$ **do**
  $z = \arg\min_{||z - x_i||_p \leq r} y_i \langle w, z \rangle$ {finds adv. ex.}
  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 $f_{w,0}$

Figure 2. An algorithm combining adversarial training with the perceptron algorithm. For each (x_i, y_i), we first attack it, to get z. If z is labeled incorrectly, we do a perceptron update using z.

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.

4. Upper Bounds

In the previous section, we showed that for any algorithm, there is some distribution $\mathcal{D} \in \mathcal{F}_{r,\rho}$ 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 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 $\mathcal{F}_{r,\rho}$ 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 $\mathcal{D}$ is a linearly r-separated distribution, then $\mathcal{D}$ must also be linearly separable. As earlier, let $S^+, S^- \subset \mathbb{R}^d$ denote the positively and negatively labeled examples from $\mathcal{D}$. We now define

$$S_r^+ = \cup_{s \in S^+} B_p(s, r) \text{ and } S_r^- = \cup_{s \in S^-} B_p(s, r). \quad (1)$$

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

Definition 17. *Let $\mathcal{D}$ be a linearly r-separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let S_r^+ and S_r^- be as above. Then $\mathcal{D}$ 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, $\mathcal{L}_r(f_{w,b}, \mathcal{D}) = 0$.

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 $\gamma_r(\mathcal{D})$ denote the margin of $\mathcal{D}$, 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 18. *Let $\mathcal{D}$ be a distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then the **robust aspect ratio** of $\mathcal{D}$, $\rho_r(\mathcal{D})$ is defined as*

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

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, $\rho(\mathcal{D})$, characterized the sample complexity for standard classification, the robust aspect ratio, $\rho_r(\mathcal{D})$ will characterize the sample complexity for robust learning. To do so, we present a perceptron-inspired algorithm (Algorithm 1) for learning a robust classifier on r-separated data with robust aspect ratio ρ_r.

The basic idea behind Algorithm 1 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 1 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 19. *Let $\mathcal{D}$ be a distribution with robust aspect ratio $\rho_r(\mathcal{D})$. Then for any $n > 0$, we have*

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

where A_S denotes the classifier learned by Algorithm 1 from training data S.

Observe that this expected loss is still larger than the expected standard loss in Theorem 10 as $\rho_r(\mathcal{D}) > \rho(\mathcal{D})$ for any $\mathcal{D}$. We also note that this result is not contradictory with our lower bound; there exist distributions $\mathcal{D} \in \mathcal{F}_{r,\rho}$ such that $\gamma_r(\mathcal{D}) = 0$, and these are precisely the distributions for which our lower bounds hold.

4.1. Generalization to Kernel Classifiers

Algorithm 1 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 D

Definition 20. *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 1 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).

Algorithm 2 Adversarial-Kernel-Perceptron

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

Figure 3. A kernel version of Algorithm 1. We replace the perceptron update step with a kernel-perceptron update step.

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).

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, $\rho_r(\mathcal{D})$, 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 D.

Theorem 21. *Let $\mathcal{D}$ be a distribution with robust K-aspect ratio $\rho^K_r(\mathcal{D})$. Then for any $n > 0$, we have*

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

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

This result indicates that for small values of $\rho^k_r(\mathcal{D})$, 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.

References

Ashtiani, H., Pathak, V., and Urner, R. Black-box certification and learning under adversarial perturbations. *CoRR*, abs/2006.16520, 2020. URL https://arxiv.org/abs/2006.16520.

Attias, I., Kontorovich, A., and Mansour, Y. Improved generalization bounds for robust learning. In Garivier, A. and Kale, S. (eds.), *Algorithmic Learning Theory, ALT 2019, 22-24 March 2019, Chicago, Illinois, USA*, volume 98 of *Proceedings of Machine Learning Research*, pp. 162–183. PMLR, 2019.

Bhagoji, A. N., Cullina, D., and Mittal, P. Lower bounds on adversarial robustness from optimal transport. In Wallach, H. M., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. B., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, 8-14 December 2019, Vancouver, BC, Canada*, pp. 7496–7508, 2019.

Bhattacharjee, R. and Chaudhuri, K. When are non-parametric methods robust? *CoRR*, abs/2003.06121, 2020. URL https://arxiv.org/abs/2003.06121.

Carlini, N. and Wagner, D. A. 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*, pp. 39–57, 2017.

Cullina, D., Bhagoji, A. N., and Mittal, P. Pac-learning in the presence of adversaries. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 31, pp. 230–241. Curran Associates, Inc., 2018.

Dan, C., Wei, Y., and Ravikumar, P. Sharp statistical guarantees for adversarially robust gaussian classification. *CoRR*, abs/2006.16384, 2020. URL https://arxiv.org/abs/2006.16384.

Diakonikolas, I., Kane, D. M., and Manurangsi, P. The complexity of adversarially robust proper learning of halfspaces with agnostic noise. *CoRR*, abs/2007.15220, 2020. URL https://arxiv.org/abs/2007.15220.

Dobriban, E., Hassani, H., Hong, D., and Robey, A. Provable tradeoffs in adversarially robust classification. *CoRR*, abs/2006.05161, 2020. URL https://arxiv.org/abs/2006.05161.

Freund, Y. and Schapire, R. E. Large margin classification using the perceptron algorithm. *Mach. Learn.*, 37(3): 277–296, 1999.

Hein, M. and Andriushchenko, M. Formal guarantees on the robustness of a classifier against adversarial manipulation. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30*, pp. 2266–2276. Curran Associates, Inc., 2017.

Katz, G., Barrett, C. W., Dill, D. L., Julian, K., and Kochenderfer, M. J. 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.*, pp. 19–26, 2017.

Khim, J. and Loh, P. Adversarial risk bounds for binary classification via function transformation. *CoRR*, abs/1810.09519, 2018. URL http://arxiv.org/abs/1810.09519.

Liu, Y., Chen, X., Liu, C., and Song, D. 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.

Montasser, O., Hanneke, S., and Srebro, N. VC classes are adversarially robustly learnable, but only improperly. In Beygelzimer, A. and Hsu, D. (eds.), *Conference on Learning Theory, COLT 2019, 25-28 June 2019, Phoenix, AZ, USA*, volume 99 of *Proceedings of Machine Learning Research*, pp. 2512–2530. PMLR, 2019.

Papernot, N., McDaniel, P. D., Jha, S., Fredrikson, M., Celik, Z. B., and Swami, A. 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*, pp. 372–387, 2016a.

Papernot, N., McDaniel, P. D., Wu, X., Jha, S., and Swami, A. 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*, pp. 582–597, 2016b.

Papernot, N., McDaniel, P. D., Goodfellow, I. J., Jha, S., Celik, Z. B., and Swami, A. Practical black-box attacks against deep learning systems using adversarial examples. *ASIACCS*, 2017.

Raghunathan, A., Steinhardt, J., and Liang, P. 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.

Raghunathan, A., Xie, S. M., Yang, F., Duchi, J. C., and Liang, P. Understanding and mitigating the

tradeoff between robustness and accuracy. *CoRR*, abs/2002.10716, 2020. URL https://arxiv.org/abs/2002.10716.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In Bengio, S., Wallach, H. M., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, 3-8 December 2018, Montréal, Canada*, pp. 5019–5031, 2018.

Sinha, A., Namkoong, H., and Duchi, J. C. 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.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. 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.

Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., and Madry, A. 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.

Vapnik, V. N. *Statistical Learning Theory*. Wiley-Interscience, 1998.

Wang, Y., Jha, S., and Chaudhuri, K. 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*, pp. 5120–5129, 2018.

Yang, Y., Rashtchian, C., Wang, Y., and Chaudhuri, K. Adversarial examples for non-parametric methods: Attacks, defenses and large sample limits. *CoRR*, abs/1906.03310, 2019. URL http://arxiv.org/abs/1906.03310.

Yang, Y.-Y., Rashtchian, C., Zhang, H., Salakhutdinov, R., and Chaudhuri, K. A closer look at accuracy vs. robustness, 2020.

Yin, D., Ramchandran, K., and Bartlett, P. L. Rademacher complexity for adversarially robust generalization. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *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*, pp. 7085–7094. PMLR, 2019. URL http://proceedings.mlr.press/v97/yin19b.html.

A. Expanded summary of (Dan et al., 2020)

In this section, we derive the formulation of Theorem 16 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 16 can be

Recall, that (Dan et al., 2020) consider the setting in which the data distribution $\mathcal{D}_{\mu,\Sigma}$ can be characterized as a pair of Gaussians in $\mathbb{R}^d$, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(-\mu, \Sigma)$, 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 $\mathcal{D}_{\mu,\Sigma}$,
   - 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 $\mathcal{D}_{\mu,\Sigma}$ 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 $\mathcal{D}_{\mu,\Sigma}$, 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 16. As before, let us define L_{std} and L_{rob} as the best possible standard and robust losses for $\mathcal{D}_{\mu,\Sigma}$ 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, $\mathcal{D}_{\mu,\Sigma}$ such that the expected robust loss is at least $\Omega(L_{rob} \frac{d}{n})$.

3. For any distribution $\mathcal{D}_{\mu,\Sigma}$, 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 16.

A.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 16, 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 14). Thus, considering Gaussians that appear linearly r-separated does not generalize to the general, linearly r-separated case.

B. Proof of Theorem 14

We begin by broadly outlining our proof of Theorem 14. Let Π be a probability distribution over $\mathcal{F}_{r,\rho}$, and let A be a learning algorithm that returns a linear classifier.

1. Sample $\mathcal{D} \sim \Pi$.

2. Sample $S \sim \mathcal{D}^n$.

3. Learn the classifier A_S using algorithm A and training sample S.

4. Evaluate A_S on $\mathcal{D}$. That is, compute $\mathcal{L}_r(A_S, \mathcal{D})$.

The basic idea of our proof is to show that for an appropriate choice of Π, the overall expected loss of this procedure, $\mathcal{L}_r(A_S, \mathcal{D})$, satisfies

$$\mathbb{E}_{D\sim\Pi}[\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] \geq \Omega(\frac{d}{n}).$$

Our primary method for doing this is switching expectations. In particular, observe that

$$\mathbb{E}_{D\sim\Pi}[\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] = \mathbb{E}_{S\sim\Sigma}[\mathbb{E}_{\mathcal{D}\sim\Pi|S}[\mathcal{L}_r(A_S, \mathcal{D})]],$$

where Σ denotes the distribution over all S obtained from first sampling $\mathcal{D} \sim \Pi$ and then sampling $S \sim \mathcal{D}^n$, and $\Pi|S$ denotes the posterior distribution of $\mathcal{D}$ after observing S. It then suffices to bound the quantity $\mathbb{E}_{\mathcal{D}\sim\Pi|S}[\mathcal{L}_r(A_S, \mathcal{D})]$, 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 B.1, we construct the distribution Π, and prove several important properties about it.

2. In section B.2, we show that the desired property of Π holds, by bounding $\mathbb{E}_{\mathcal{D}\sim\Pi|S}[\mathcal{L}_r(A_S, \mathcal{D})]$.

B.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 B.1.1, we define the distribution $\mathcal{D}_a$, characterized by parameter $a \in [0,1]^d$. This forms the basis for constructing Π, which will comprise of distributions $\mathcal{D}_a$ for certain choices of a. We also show that $\mathcal{D}_a$ is linearly r-separated.
- In section B.1.2, we define the constant Δ, which will be essential for specifying which values of parameter a are permissible.
- In section B.1.3, 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 B.1.4, we finally put together the previous 3 sections and construct Π. We also show that any $\mathcal{D}_a \sim \Pi$ satisfies $\rho(\mathcal{D}_a) \leq C$.

B.1.1. Defining $\mathcal{D}_a$

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 22. *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 $\mathcal{D}_a$, we first define the concept of a line segment in $\mathbb{R}^d$.

Definition 23. *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 $\mathcal{D}_a$.

Definition 24. *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 $\mathcal{D}_a$ for a for which this holds). Let S^- and S^+ be two sets of d disjoint line segments (as defined in Definition 23) 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.*
- *Y is -1 if $X \in \cup S^-$ and $+1$ if $X \in \cup S^+$.*

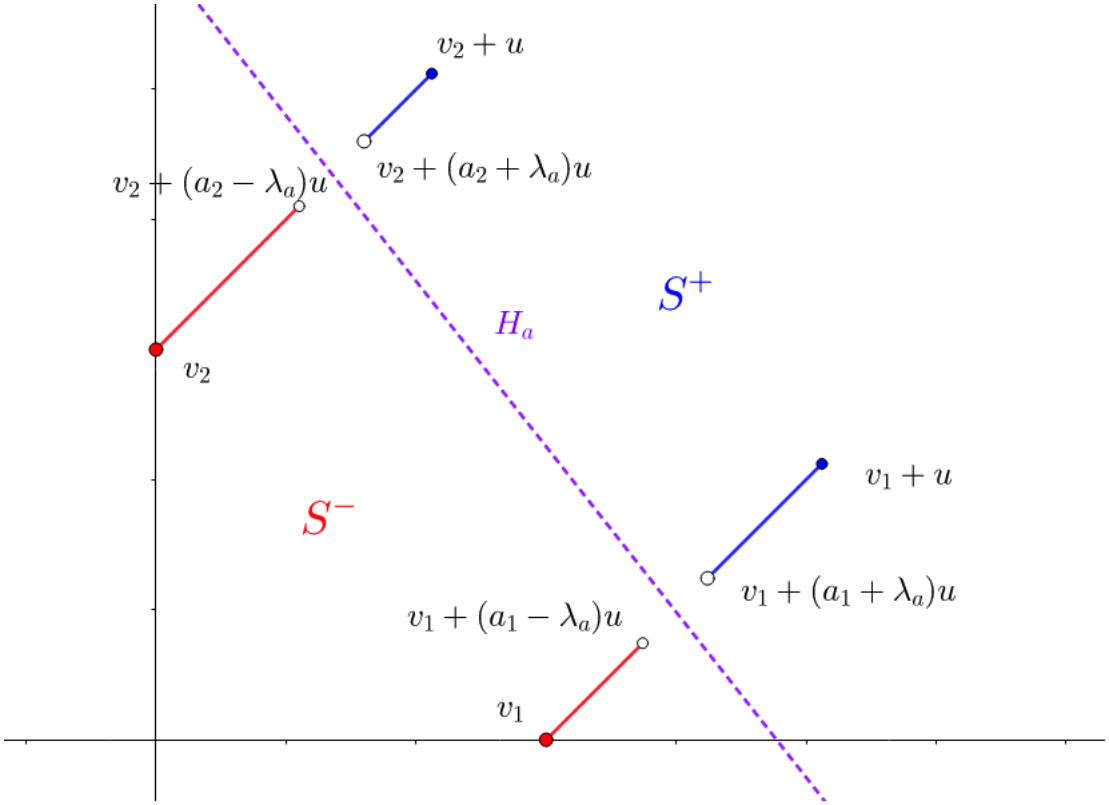

Figure 4. An illustration of $\mathcal{D}_a$ in two dimensions. S^- is shown in red, and S^+ is shown in blue. The decision boundary, H_a, of the optimal linear classifier, $f_{w^a,1}$, is shown in purple.

We include an example of such a distribution in Figure 4. Next, we explicitly compute a linear classifier that linearly r-separates $\mathcal{D}_a$.

Definition 25. *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 26. *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 $1 \le i \le d$.*

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 $\mathcal{D}_a$ is linearly r-separated.

Lemma 27. *$\mathcal{D}_a$ 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 26, H_a is the decision boundary of $f_{w_a,1}$. Referring to Figure 4, 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 $\mathcal{D}_a$. 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 $\mathcal{D}_a$ 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

$$\max_{||z-x_i||_p \le D}\langle z - x_i, w^a\rangle = 1 - \langle x_i, w^a\rangle.$$

By Lemma 26 , $\langle w^a, u\rangle = \frac{d}{\sqrt{d}+d\overline{a}}$ 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. □

B.1.2. Defining Δ

Now that we have defined $\mathcal{D}_a$, we turn our attention to defining Π, which requires us to specify a distribution over valid choices of a. In particular, although $\mathcal{D}_a$ 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 28. *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 22.

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 29. *Let Δ be any constant for which Lemma 28 holds. In particular, Δ only depends on ℓ_p, the robustness norm, and d, the dimension.*

B.1.3. 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 30. *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 31. *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 $\epsilon > 0$ be such that $x - \epsilon \in I$ and $y + \epsilon \leq z - \epsilon$. Then there exist unique $\delta, \gamma > 0$ such that the following hold:*

$$\delta + \gamma = \epsilon,$$

$$\Phi(x - \delta) + \Phi(y + \epsilon) + \Phi(z - \gamma) = \Phi(x) + \Phi(y) + \Phi(z)$$

Proof. Fix any ϵ satisfying the desired conditions, and define $\Theta : [0, \epsilon] \to \mathbb{R}$ as $\Theta(t) = \Phi(x - t) + \Phi(y + \epsilon) + \Phi(z + t - \epsilon)$. Then, utilizing the definition of Φ_s from Lemma 30, we see that

$$\Theta(t) = \Phi_{\frac{x+z-\epsilon}{2}}(\frac{z - x - \epsilon}{2} + t) + \Phi(y + \epsilon).$$

By Lemma 30, it follows that Θ is strictly increasing in t, and since Φ is continuous, so is Θ. Next, we bound $\Theta(0)$ and $\Theta(\epsilon)$ to put us in the configuration to apply the intermediate value theorem. To bound $\Theta(0)$, we have

$$\begin{aligned}\Theta(0) &= \Phi(x) + \Phi(y+\epsilon) + \Phi(z-\epsilon)\\ &= \Phi(x) + \Phi_{\frac{y+z}{2}}(\frac{z-y}{2} - \epsilon)\\ &< \Phi(x) + \Phi_{\frac{y+z}{2}}(\frac{z-y}{2})\\ &= \Phi(x) + \Phi(y) + \Phi(z),\end{aligned}$$

and to bound $\Theta(\epsilon)$, we have

$$\begin{aligned}\Theta(\epsilon) &= \Phi(x-\epsilon) + \Phi(y+\epsilon) + \Phi(z)\\ &= \Phi_{\frac{x+y}{2}}(\frac{y-x}{2} + \epsilon) + \Phi(z)\\ &> \Phi_{\frac{x+y}{2}}(\frac{y-x}{2}) + \Phi(z)\\ &= \Phi(x) + \Phi(y) + \Phi(z).\end{aligned}$$

Together, these equations imply $\Theta(0) < \Phi(x) + \Phi(y) + \Phi(z) < \Theta(\epsilon)$. Since Θ is strictly increasing and continuous, there exists a unique $\delta \in [0, \epsilon]$ such that $\Theta(\delta) = \Phi(x) + \Phi(y) + \Phi(z)$. Setting $\gamma = \epsilon - \delta$, we see that

$$\Theta(\delta) = \Phi(x-\delta) + \Phi(y+\epsilon) + \Phi(z-\gamma) = \Phi(x) + \Phi(y) + \Phi(z),$$

as desired.

□

Next, we define a function that will be useful for simplifying notation, both in this section and subsequent ones.

Definition 32. *Let Δ be as in definition 29. 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 33. *Let Δ be as in definition 29. 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 31. 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 $0 \le t_1 < t_2 \le \frac{\Delta}{3}$, and let $t_2 - t_1 = \epsilon$. The key idea is to apply Lemma 31 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

$$-\frac{2\Delta}{3} - g_2(t_1) - \epsilon \ge -\frac{2\Delta}{3} - t_1 - \epsilon = -\frac{2\Delta}{3} - t_2 \ge -\Delta,$$

and

$$\begin{aligned} t_1 + \epsilon &= t_2 \\ &\le \frac{\Delta}{3} \\ &\le \frac{2\Delta}{3} - t_2 \\ &= \frac{2\Delta}{3} - t_1 - \epsilon \\ &\le \frac{2\Delta}{3} - g_1(t_1) - \epsilon. \end{aligned}$$

Thus ϵ satisfies the necessary conditions for Lemma 31. Since Φ is strictly convex, by Lemma 31, there exist unique $\delta, \gamma > 0$ with $\delta + \gamma = \epsilon$ such that

$$\Phi(-\frac{2\Delta}{3} - g_2(t_1) - \delta) + \Phi(t_1 + \epsilon) + \Phi(\frac{2\Delta}{3} - g_1(t_1) - \gamma) = \Phi(-\frac{2\Delta}{3} - g_2(t_1)) + \Phi(t_1) + \Phi(\frac{2\Delta}{3} - g_1(t_1)).$$

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, $t_2 - t_1 = \epsilon$, and $\delta, \gamma < \epsilon$ (since they sum to ϵ). Thus, we see that $|g_1(t_2) - g_1(t_1)| \le |t_2 - t_1|$ and $|g_2(t_2) - g_2(t_1)| \le |t_2 - t_1|$. 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 28), 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 34. *Let Δ be as defined in Definition 29. 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 33.*

B.1.4. Putting it all together: defining Π

We are now ready to define Π. For convenience, we assume d is a multiple of 3.

Definition 35. *Let Δ, g_1, and g_2 be as defined in Definitions 29 and 34. Then Π is defined as the distribution of distributions $\mathcal{D}_a$ 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 $\mathcal{D} \sim \Pi$ can be constructed by sampling a as above and setting $\mathcal{D} = \mathcal{D}_a$.

We now show that for all $\mathcal{D}_a \sim \Pi$, λ_a (Definition 24) is constant.

Lemma 36. *There exists a constant Λ such that for all $\mathcal{D}_a \sim \Pi$, $\lambda_a = \Lambda$.*

Proof. Let $\mathcal{D}_a \sim \Pi$ be arbitrary. By Lemma 33, 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 32. Next, by Corollary 33, $F(t_i, g_1(t_i), g_2(t_i)) = F(0,0,0)$ for all $1 \le i \le \frac{d}{3}$. 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 37. *We define $\Lambda = \frac{r}{R}(\frac{d}{3})^{1/q} F(0,0,0)$, where F is defined as in Definition 32.*

Next, we compute upper and lower bounds on Λ, both of which will be useful for subsequent lemmas.

Lemma 38. $\frac{1}{9} < \Lambda < \frac{1}{3}$.

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 $\mathcal{D}_a \in \Pi$, the aspect ratio (Definition 9), $\rho(\mathcal{D}_a)$, is bounded by a constant.

Lemma 39. *For all $\mathcal{D}_a \in \Pi$, we have $\rho(\mathcal{D}_a) \leq 18\sqrt{3}$.*

Proof. We first bound the ℓ_2 margin, $\gamma(\mathcal{D}_a)$ (Definition 8). Recall that the margin, $\gamma(\mathcal{D}_a)$ is described as the largest possible ℓ_2 distance from the support of $\mathcal{D}_a$ to the decision boundary of a linear classifier. Thus, we can lower bound $\gamma(\mathcal{D}_a)$ by computing the distance from the support of $\mathcal{D}_a$ to H_a, the decision boundary of $f_{w^a,1}$ (Definition 25).

By referring to Figure 4 (in Section B.1.1), 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 26, 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 36, (2) holds by Lemma 26, (3) holds by Definition 25, and (4) holds by Definition 22.

Next, observe that since $\mathcal{D}_a \sim \Pi$, 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 28, 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 38, we see that

$$\gamma(\mathcal{D}_a) \geq \frac{R\Lambda}{2} \geq \frac{R}{18}.$$

Next, to bound the aspect ratio, $\rho(\mathcal{D}_a)$, we must also bound the ℓ_2 diameter of $\mathcal{D}_a$. However, the ℓ_s diameter of $\mathcal{D}_a$ is $R\sqrt{3}$, since it is the distance from $v_i + u$ to v_j for $i \neq j$. Thus, it follows that

$$\rho(\mathcal{D}_a) = \frac{diam_2(\mathcal{D}_a)}{\gamma(\mathcal{D}_a)} \leq \frac{R\sqrt{3}}{R/18} = 18\sqrt{3},$$

as desired. □

Note that a tighter analysis (and selection of Δ) can give a smaller bound for $\rho(\mathcal{D}_a)$, but the most important fact is that $\rho(\mathcal{D}_a) = O(1)$.

B.2. Bounding the expected robust loss

In this section, we finally prove our lower bound, Theorem 14. This will require a few important steps, which we have separated into the following subsections.

- In section B.2.1, we give a useful lower bound for the loss $\mathcal{L}_r(f, \mathcal{D}_a)$ where f is an arbitrary linear classifier.
- In section B.2.2, we give an explicit computation for the posterior distribution $\Pi|S$ where $S \sim \mathcal{D}_a^n$ is the observed training sample.
- Finally, in section B.2.3, we present the proof of Theorem 14.

B.2.1. Bounding the loss $\mathcal{L}_r(f, \mathcal{D}_a)$

In this section, we find a lower bound on the loss $\mathcal{L}_r(f, \mathcal{D}_a)$ 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 25. 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 $\mathcal{D}_a$ 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 $\mathcal{L}_r(f_{w^b,1}, \mathcal{D}_a)$.

Lemma 40. *Fix any $\mathcal{D}_a \in \Pi$, and let $b \in [0,1]^d$ be arbitrary. Let w^b be the vector defined as in Definition 25, and $\lambda_b = \frac{r}{R} f_q(b)$ where f is as defined in Definition 22. Then*

$$\mathcal{L}_r(f_{w^b,1}, \mathcal{D}_a) \geq \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda}.$$

Proof. By Lemma 27, $f_{w^b,1}$ precisely r-separates $\mathcal{D}_b$. 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 $\mathcal{D}_a$, we find that

$$\begin{aligned}
\mathcal{L}_r(f_{w^b,1}, \mathcal{D}_a) &\geq \frac{[d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|]||u||_2}{\sum_1^d |||[v_i, v_i + (a_i - \lambda_a)u) + (v_i + (a_i + \lambda_a)u, v_i + u]||_2} \\
&= \frac{[d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|]||u||_2}{\sum_1^d ||u_2||(1 - 2\lambda_a)} \\
&= \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d(1 - 2\lambda_a)} \\
&= \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda},
\end{aligned}$$

with the last equality holding since by Lemma 36, $\lambda_a = \Lambda$. □

Lemma 41. *For all $\mathcal{D}_a \in \Pi$ 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 28, 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

$$\lambda_b = \frac{r}{R} f_q(b) = \frac{r}{R}\sqrt[q]{\sum_1^d |\frac{1}{\sqrt{d}} + \bar{b} - b_i|^q}.$$

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

$$b_i \leftarrow \begin{cases} \begin{cases} (\frac{1}{2} - \Delta)(\frac{\alpha - 1}{\alpha}) + \frac{b_i}{\alpha} & 1 \le i \le l \\ \frac{1}{2} - \Delta & k+1 \le i \le d \end{cases} & \alpha \ge 1 \\ \begin{cases} \frac{1}{2} - \Delta & 1 \le i \le l \\ \frac{1}{2} - \Delta + \frac{\beta}{d-l}(1 - \alpha) & l+1 \le i \le d \end{cases} & \alpha < 1 \end{cases}.$$

The justification for this step is analogous to 3a.

Step 4: We only run this step if $\alpha < 1$. Observe that if $\alpha \ge 1$, 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 $b_i \ge a_i$ for all i, or $b_i \le a_i$ 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 $1 \le i \le d\Delta$, 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 42. *For all $\mathcal{D}_a \in \Pi$ and $b \in [0,1]^d$,*

$$\mathcal{L}_r(f_{w^b,1}, \mathcal{D}_a) \geq \frac{1}{2d}\sum_1^d |a_i - b_i|.$$

Proof. We have that

$$\begin{aligned}
\mathcal{L}_r(f_{w^b,1}, \mathcal{D}_a) &\overset{(a)}{\geq} \frac{d(\lambda_b - \lambda_a) + \sum_1^d |a_i - b_i|}{d - 2d\Lambda} \\
&\geq \frac{\sum_1^d |a_i - b_i| - d(\lambda_a - \lambda_b)+}{d} \\
&\overset{(b)}{\geq} \frac{\sum_1^d |a_i - b_i| - \frac{1}{2}\sum_1^d |a_i - b_i|}{d} \\
&= \frac{1}{2d}\sum_1^d |a_i - b_i|,
\end{aligned}$$

where (a) holds by Lemma 40 and (b) holds by Lemma 41. □

B.2.2. Computing the posterior distribution, $\Pi|S$

Recall that our ultimate goal is to show that

$$\mathbb{E}_{\mathcal{D}\sim\Pi}[\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] \geq \Omega(\frac{d}{n}),$$

where A denotes any learning algorithm returning a linear classifier. The main idea for showing this is to "switch expectations" and realize that

$$\mathbb{E}_{\mathcal{D}\sim\Pi}[\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] = \mathbb{E}_{S\sim\Sigma}[\mathbb{E}_{\mathcal{D}\sim\Pi|S}[\mathcal{L}_r(A_S, \mathcal{D})]],$$

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 35) that $\mathcal{D}_a \sim \Pi$ 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 43. *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 35. 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 $\mathcal{D}_a^n$. In particular, for any measurable region of points $R \subseteq (\mathbb{R}^d \times \{\pm 1\})^n$, we have that

$$\mathbb{P}_{S \sim \mathcal{D}_a^n}[S \in R] = \int_{x \in R} L(x|t)dx.$$

Lemma 44. *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 37 and $L(S|t)$ is as defined in Definition 43.

Proof. Let $\mathcal{D}_a$ be an arbitrary distribution in Π. Observe that $\mathcal{D}_a$ 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 $\mathcal{D}_a$, then the likelihood becomes 0, since the likelihood of observing some point not in the support of $\mathcal{D}_a$ is 0. □

Definition 45. *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 $\mathcal{D} \sim \Pi$.

Lemma 46. *Fix $n > 0$. For all $\mathcal{D} \sim \Pi$ and $S \sim \mathcal{D}^n$, 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 $S \sim \mathcal{D}^n$, 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 35. 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 $\mathcal{D}_a$ (Definition 24), 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 $S \subseteq supp(\mathcal{D}_a)$, 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 47. *For any $S \sim \mathcal{D}$ where $\mathcal{D} \sim \Pi$, let I_i^S be defined as in Lemma 46 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 44, 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 46, we see that $P_S = \prod_1^{1 \leq i \leq d/3} I_i^S$. 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 48. *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 $1 \leq k \leq d/3$, the expected length (taken over $\mathcal{D}_a \sim \Pi$ and $S \sim \mathcal{D}_a^n$) of the interval I_k^S is at least $\Omega(\frac{d}{n})$. That is,*

$$\mathbb{E}_{\mathcal{D}_a \sim \Pi} \mathbb{E}_{S \sim \mathcal{D}_a^n}[\ell(I_k^S)]] \geq \Omega(\frac{d}{n}).$$

Proof. Fix any $\mathcal{D}_{a^*} \sim \Pi$, and let t^* denote the value of t used to generate a (as in Definition 35). We will show that $\mathbb{E}_{S \sim \mathcal{D}_{a^*}^n}[\ell(I_k^S)]] \geq \Omega(\frac{d}{n})$, for all $1 \leq k \leq d/3$. We begin by explicitly computing the interval I_k^S.

Fix $1 \leq k \leq d/3$. 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 46 that for $1 \leq i \leq d$, 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 $1 \leq i \leq d$.

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 $1 \leq k \leq d/3$. Note that g_1, g_2 are the functions defined in Definition 34.

As we argued in the proof of Lemma 46, 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 46, 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 $\mathcal{D}_{a^*}$, 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 $\mathbb{E}_{S \sim \mathcal{D}_{a^*}}[\min(\alpha, \beta, \tau)] \geq \Omega(\frac{d}{n})$.

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 $\mathbb{E}_{S \sim \mathcal{D}_{a^*}}[\min(\alpha, \beta, tau)] \geq \Omega(\frac{d}{n})$, as desired. □

B.2.3. PUTTING IT ALL TOGETHER, THE PROOF

We prove the following key lemma, which directly implies Theorem 14.

Lemma 49. *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 $S \sim \mathcal{D}$. Then it follows that*

$$\mathbb{E}_{\mathcal{D}\sim\Pi}\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(M_S, \mathcal{D})]] \geq \Omega(\frac{d}{n}).$$

Proof. Let $\mathcal{F}_n$ denote the distribution over $(\mathbb{R}^d \times \{\pm 1\})^n$ defined as the composition $\mathcal{D} \sim \Pi$ and $S \sim \mathcal{D}^n$. That is, $S \sim \mathcal{F}_n$ follows the same distribution as $\mathcal{D} \sim \Pi, S \sim \mathcal{D}^n$. Then we can write the expectation above as

$$\mathbb{E}_{\mathcal{D}\sim\Pi}\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] = \mathbb{E}_{S\sim\mathcal{F}_n}\mathbb{E}_{\mathcal{D}\sim(\Pi|S)}[\mathcal{L}_r(M_S, \mathcal{D})]],$$

where $\Pi|S$ denotes the posterior distribution of $\mathcal{D}$ conditioned on observing S. First, fix any such S. We will bound $\mathbb{E}_{\mathcal{D}\sim(\Pi|S)}[\mathcal{L}_r(M_S, \mathcal{D})]$. First, by reparametrizing in terms of $t \in [0, \frac{\Delta}{3}]^{d/3}$ and applying Corollary 47, we have that

$$\mathbb{E}_{D\sim(\Pi|S)}[\mathcal{L}_r(M_S, \mathcal{D})] = \mathbb{E}_{t_1\sim\mathbb{U}(I_1^S)}[\ldots[\mathbb{E}_{t_n\sim\mathbb{U}(I_{d/3})}[\mathcal{L}_r(M_S, \mathcal{D}_a)]\ldots],$$

where $I_1^S, I_2^S, \ldots, I_{d/3}^S \subset [0, \frac{\Delta}{3}]$ are the intervals defined in Lemma 46, and a is defined as in Definition 35.

Next, let $b \in [0,1]^d$ be such that $M_S = f_{w^b,1}$, where w^b is defined as in Definition 25. Then it follows from Corollary 42 that

$$\begin{aligned}\mathcal{L}_r(M_S, \mathcal{D}_a)] &\geq \frac{1}{20d}\sum_1^d |a_i - b_i| \\ &\geq \frac{1}{20d}\sum_1^{d/3} |\frac{1}{2} + t_i - b_i|\end{aligned}$$

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 $t_1, t_2, \ldots, t_{d/3}$. 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

$$E_{t_1\sim\mathbb{U}(I_1^S)}[\ldots[\mathbb{E}_{t_n\sim\mathbb{U}(I_{d/3}^S)}[\mathcal{L}_r(M_S, \mathcal{D}_a)]\ldots] \geq \frac{1}{80d}\sum_{i=1}^{d/3}\ell(I_i^S).$$

Finally, by taking expectations over $S \sim \mathcal{F}_n$, we see that

$$\begin{aligned}\mathbb{E}_{\mathcal{D}\sim\Pi}\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(A_S, \mathcal{D})]] &= \mathbb{E}_{S\sim\mathcal{F}_n}\mathbb{E}_{\mathcal{D}\sim(\Pi|S)}[\mathcal{L}_r(M_S, \mathcal{D})]] \\ &\geq \mathbb{E}_{S\sim\mathcal{F}_n}\frac{1}{80d}\sum_{i=1}^{d/3}\ell(I_i^S) \\ &= \frac{1}{80d}\sum_1^{d/3}\mathbb{E}_{S\sim\mathcal{F}}[\ell(I_i^S)] \\ &= \frac{1}{80d}\sum_1^{d/3}\mathbb{E}_{\mathcal{D}\sim\Pi}\mathbb{E}_{S\sim\mathcal{D}^n}[\ell(I_i^S)] \\ &\geq \frac{1}{80d}\sum_1^{d/3}\Omega(\frac{d}{n}) = \Omega(\frac{d}{n}),\end{aligned}$$

where the last step follows from Lemma 48. □

Finally, we can prove Theorem 14.

Proof. (Theorem 14). First, by Lemmas 27 and 39, we see that $\Pi \subseteq \mathcal{F}_{r,\rho}$ (provided $\rho > 10$). Next, by Lemma 49, for any n there must exists some $\mathcal{D} \sim \Pi$ such that $\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(M_S, \mathcal{D})] \geq \Omega(\frac{d}{n})$. Thus selecting this distribution suffices. This concludes the proof. □

C. Proofs for Algorithm 1

This section is divided into 2 parts. In section C.1, we show that for the case in which our data distribution $\mathcal{D}$ is linearly r-separated by some hyperplane through the origin, the desired error bound holds. That is, we prove Theorem 19 under this assumption.

Next, in section C.2, we show how to generalize Algorithm 1 to arbitrary linearly r-separated distributions, and subsequently prove Theorem 19 in the general case.

C.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 $\mathcal{D}$. We let S_r^+ and S_r^- be defined as in section 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
   - $\mathcal{L}_r(f_{u,0}, \mathcal{D}) = 0$, 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 $(x, y) \sim \mathcal{D}$, and $x' \in B_p(x, r)$. This is because u is a unit vector.

Next, before analyzing Algorithm 1, 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 3 Modified-Adversarial-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathcal{D}^n$,
$w \leftarrow 0$
$k \sim \mathbb{U}(\{0, 1, 2, \ldots, n\})$
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 3 satisfies the guarantees of Theorem 51. We begin with the following, key lemma.

Lemma 50. *Under the assumptions above about $\mathcal{D}$, Algorithm 3 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 51. *Let $\mathcal{D}$ be a distribution with the assumptions above. For any $S \sim \mathcal{D}^n$, let f_S denote the classifier learned by Algorithm 3. Then*

$$\mathbb{E}_{S\sim\mathcal{D}^n} \mathcal{L}_r(f_S, \mathcal{D}) \leq \frac{R^2}{\gamma_r^2(n+1)}.$$

This Theorem directly follows from the classic online to offline result (Theorem 3 of (Freund & Schapire, 1999)). For completeness, we include a proof in our context.

Proof. Fix any n and consider running Algorithm 3 on $S \sim \mathcal{D}^n$. 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

$$\mathbb{E}_{S\sim\mathcal{D}^n} L^* = \frac{1}{n+1}\sum_{t=0}^{n} \mathbb{E}_{S\sim\mathcal{D}^n}[L^*|k=t] = \frac{1}{n+1}\sum_{t=0}^{n} \mathbb{E}_{S\sim\mathcal{D}^n}[L_t].$$

Next, let $T \sim \mathcal{D}^{n+1}$ 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 3 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 $\mathbb{E}_{T\sim\mathcal{D}^{n+1}}[X_t] = \mathbb{E}_{S\sim\mathcal{D}^n}[L_{t-1}]$. 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 $\mathcal{D}$. 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 50. By combining these two observations, we see that

$$\begin{aligned}
\mathbb{E}_{S\sim\mathcal{D}^n}[L^*] &= \frac{1}{n+1}\sum_{t=0}^{n} \mathbb{E}_{S\sim\mathcal{D}^n}[L_t] \\
&= \frac{1}{n+1}\sum_{t=0}^{n} \mathbb{E}_{T\sim\mathcal{D}^{n+1}}[X_{t+1}] \\
&= \frac{1}{n+1}\mathbb{E}_{T\sim\mathcal{D}^{n+1}}[\sum_{t=1}^{n+1} X_t] \\
&\leq \frac{R^2}{\gamma_r^2(n+1)},
\end{aligned}$$

as desired. □

C.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 52. *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 1, that is capable of handling any dataset, including ones that aren't separated by a hyperplane through the origin.

Algorithm 4 General-Adversarial-Perceptron

Input: $S = \{(x_1,y_1),\ldots,(x_n,y_n)\} \sim \mathcal{D}^n$,
$x_i' \leftarrow x_i - x_1$.
$R_S = diam_2(S)$
$w \leftarrow 0 \in \mathbb{R}^{d+1}$
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 1 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 53. *Let $\mathcal{D}$ be any linearly r-separated distribution, and let $S \sim \mathcal{D}^n$ 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 $(x_i,y_i) \in S$, $\min_{z \in B_p^*(x_i')} \langle u, y_i(z|R_S)\rangle \geq \frac{\gamma_r(\mathcal{D})}{\sqrt{2}}$.*
- *For all $(x_i,y_i) \in S$, $||x_i'|R_S||_2 \leq \sqrt{2} diam_2(\mathcal{D})$.*

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 $\mathcal{D}$ is r-separated, there exist w, b (with w a unit vector) such that

$$\langle w, zy\rangle \geq by + \gamma_r(\mathcal{D}),$$

for all $(x, y) \sim \mathcal{D}$ and $z \in B_p(x, r)$. Furthermore, since $x_1 = 0$, it follows that $||x||_2 \leq \text{diam}_2(\mathcal{D})$ for all $(x, y) \sim \mathcal{D}$. This immediately implies that $||x_i|R_S||_2 \leq \sqrt{\text{diam}_2(\mathcal{D})^2 + R_S^2} \leq \sqrt{2}\text{diam}_2(\mathcal{D})$, yielding the second part of the lemma.

For the first part, observe that we can rearrange the equation above, we see that

$$\langle w| - \frac{b}{R_S}, zy|R_S\rangle \geq \gamma_r(\mathcal{D}).$$

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 $\langle w, x_i\rangle \geq b + \gamma_r(\mathcal{D}) > b$ 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 53 allows us to analyze the performance of Algorithm 4. 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 51 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 53. Thus, putting this all together, Theorem 19 follows.

D. 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 54. *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 55. *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 2, is that it corresponds to running Algorithm 1 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 55 that the following algorithm is identical to Algorithm 2.

Algorithm 5 Adversarial-Kernel-Perceptron

Input: $S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \sim \mathcal{D}^n$, 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 2 and 5, we have by Theorem 55 that for all time steps t,

$$w = \sum_{(z,y)\in T} y\phi(z).$$

Therefore, to analyze the performance of Algorithm 2, it suffices to analyze Algorithm 5. However, we already have built to the tools for doing this: all of the results from Section C.1 apply to Algorithm 5 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 $\mathcal{D}$ needed for Theorem 21. We begin by first defining (K, r)-separability and K-robust margin, $\gamma_{r,K}$, the Kernel analogs of linear r-separability (Definition 12) and the robust margin (Definition 17).

Definition 56. *For any $r > 0$, a distribution $\mathcal{D}$ over $\mathbb{R}^d \times \{\pm 1\}$ is (K, r)-**separable** if there exists a kernel classifier $f^K_{S,\alpha}$ such that $\mathcal{L}_r(f^K_{S,\alpha}, \mathcal{D}) = 0$.*

To define the K-robust margin, we will once again need the sets S_r^+ and S_r^- defined in equation 1 (top right of page 7). Recall that these sets denote the positively and negatively labeled elements from $supp(\mathcal{D})$ *including* all adversarial perturbations of those points.

Definition 57. *Let $\mathcal{D}$ be a (K, r)-separable distribution over $\mathbb{R}^d \times \{\pm 1\}$. Then $\mathcal{D}$ 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. *$\mathcal{L}_r(f^K_{T,\alpha}, \mathcal{D}) = 0$.*
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_r^K from H_w inside H. That is,*

$$\inf_{x\in S_r^+\cup S_r^-} \inf_{z\in H_w} \sqrt{\langle \phi(x) - z, \phi(x) - z\rangle} = \gamma_r^K.$$

We now state the main theorem giving the performance of Algorithm 2.

Theorem 58. *Let $\mathcal{D}$ 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 $x \in S_r^+ \cup S_r^-$, $\langle \phi(x), \phi(x)\rangle \le R^2$.*
2. *$\mathcal{D}$ is K, r-separable, and has K-robust margin $\gamma_r^K > 0$.*

Then for any $S \sim D^n$, if $f^k_{T,\alpha}$ denotes the classifier learned by Algorithm 2, then

$$\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{L}_r(f^k_{T,\alpha}, \mathcal{D})] = O\left(\frac{(\gamma_r^K)^2}{R^2(n+1)}\right).$$

Proof. The key idea is to observe that Lemmas 50 and 51 both directly translate from Algorithm 4 to Algorithm 5. 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 51. □