A Hard distribution for randomized smoothing described in (3)

Consider the following data distribution. For ϵ that will be fixed later, let $S_\epsilon(0) \subseteq \mathbb{R}^d$ be a sphere of radius ϵ around 0 and $N \subseteq S_\epsilon(0)$ be a set of cardinality $e^{0.118d}$ such that for all $x, y \in N, x \neq y$ we have $||x-y||_2 \geq 1.2\epsilon$. One can show that such a set exists using bounds for the surface area of spherical caps in high dimension (see Blum et al. (2015)).

Let the binary classification task be as follows. Let the distribution $\mathcal{D}_{+1}$ for class $+1$ be such that $\text{supp}(\mathcal{D}_{+1}) = (N \cup \{0\}) + B_{0.01\epsilon}$ (where the $+$ denotes the Minkowski sum). The density function on $B_{0.01\epsilon}(0)$ is $e^{0.108d}$ times larger than the one on $B_{0.01\epsilon}(u)$ for every $u \in N$. Now let $\mathcal{D}_{-1}$ be such that $\text{supp}(\mathcal{D}_{-1}) \cap \text{supp}(\mathcal{D}_{+1}) = \emptyset$ and each class has probability $1/2$. Now assume that the points that classifier f misclassifies are exactly points in $B_{0.01\epsilon}(0)$. Then the standard error of f is at most $e^{-0.01d}$. Now let $\epsilon := \sqrt{(d\sigma)/10}$. One can verify that when g is computed according to (1) then for all $x \in N + B_{0.01\epsilon}$ we have $g(x) = -1$, which means that g misclassifies all points from $N + B_{0.01\epsilon}$. So the standard error of g is at least 25%. This means that the error of g is $e^{\Theta(d)}$ times larger than the error of f!

Remark 2. *One might argue that this example was crafted artificially and that in the "real world" we can choose σ depending on the data. However it is possible to construct examples such that for* any *reasonable choice of σ a dynamic similar to the one presented above occurs. The idea is to put a collection of the above configurations at different scales and far from each other.*

B Generalization of definitions to nonseparable learning tasks

Definition 2a. For a binary classification task and a classifier $f : \mathbb{R}^d \to \{-1, 1\}$ we define **R**isk as

$$R(f) := \int p_X(x) \sum_{y \in \{-1,1\}} \mathbb{P}_{Y|X}(y|x) \mathbb{1}_{\{f(x)=y\}} dx.$$

Definition 3a. For a binary classification task, a classifier $f : \mathbb{R}^d \to \{-1, 1\}$, and $\epsilon \geq 0$ we define **A**dversarial **R**isk as

$$AR(f, \epsilon) := \int p_X(x) g(f, x, \epsilon) dx,$$

where

$$g(f, x, \epsilon) := \begin{cases} \mathbb{P}_{Y|X}(-1 \mid x), & B_\epsilon(x) \subseteq M_1(f), \\ \mathbb{P}_{Y|X}(1 \mid x), & B_\epsilon(x) \subseteq M_{-1}(f), \\ 1, & \text{otherwise}, \end{cases}$$

where $M_y = f^{-1}(\{y\})$, $y \in \{-1, 1\}$. We also introduce the notation:

$$AR(\epsilon) := \inf_f AR(f, \epsilon),$$

to denote the optimal classification error for that classification task with a given ϵ.

Note that this definition assumes that the adversary, apart from x, has also access to the label y. In other words, we prove bounds with respect to a strong adversary.

Definition 4a (Separation function). For a binary classification task we define the separation function $S(\epsilon)$ as follows:

$$S(\epsilon) := \inf_{\substack{E_{-1}, E_1 \subseteq \mathbb{R}^d \\ d(\mathbb{R}^d \setminus E_{-1}, \mathbb{R}^d \setminus E_1) \geq \epsilon}} \sum_{y \in \{-1,1\}} \int_{x \in E_y} p_X(x) \mathbb{P}_{Y|X}(y \mid x) dx.$$

For a given $\epsilon > 0$ this function returns the minimum probability mass that needs to be removed so that the classes are separated by an ϵ-margin.

Lemma 7. *For all binary classification tasks and all $\epsilon \geq 0$ we have that:*

$$AR(\epsilon) = S(2\epsilon).$$

Proof. First we prove that $AR(\epsilon) \leq S(2\epsilon)$. Let E_{-1} and E_1 be the minimizer sets from the definition of $S(2\epsilon)$. Let $f(x) := -1$ if $d(x, \mathbb{R}^d \setminus E_{-1}) \leq \epsilon$ and $f(x) := 1$ otherwise. Then observe that for all $x \in (\mathbb{R}^d \setminus E_{-1})$, $B_\epsilon(x) \subseteq M_{-1}(f)$ and for all $x \in (\mathbb{R}^d \setminus E_1)$, $B_\epsilon(x) \subseteq M_1(f)$. Hence $AR(\epsilon) \leq S(2\epsilon)$.

Now we prove that $AR(\epsilon) \geq S(2\epsilon)$. Let f be a classifier with $AR(f, \epsilon) = r$. Let E_{-1} be the set of all points $x \in \mathbb{R}^d$ so that $B_\epsilon(x) \not\subseteq M_{-1}(f)$ and let E_1 be the set of all points $x \in \mathbb{R}^d$ so that $B_\epsilon(x) \not\subseteq M_1(f)$. It follows that

$$d(\mathbb{R}^d \setminus E_{-1}, \mathbb{R}^d \setminus E_1) \geq 2\epsilon. \quad (13)$$

But now note that for this choice of sets E_{-1} and E_1,

$$\sum_{y \in \{-1,1\}} \int_{x \in E_y} p_X(x) \mathbb{P}_{Y|X}(y \mid x) dx = r = AR(f, \epsilon).$$

Hence, for $S(2\epsilon)$, being defined as the infimum over all choices of sets E_{-1} and E_1 which fulfill (13), we have $S(2\epsilon) \leq r = AR(f, \epsilon)$. □

C Running time discussion

Let us now analyze the running times of Algorithm 1 as a function of the used partition as well as the method of estimating g. In the stated bounds we will assume that each evaluation of f takes time t.

C.0.1 Cube partition

Scheme B: First let's analyze the performance of Cube partitions together with assumption (9). To evaluate $\hat{g}(x)$ we need to locate a cube to which x belongs to and smooth f over that cube. Smoothing is approximated by a sample mean and as argued before $O(\log(Q))$ samples suffice. So in the end the running time per query is $O(t \cdot \log(Q))$. If we store (using hashing techniques) previous function evaluations then the query time drops to $O(1)$ for queries from cubes that were already queried before.

Scheme A: If we use (4) instead of (9) then we first perform a preprocessing step in which we sample a set U of unlabeled samples of size (6). Then using standard hashing techniques we can create a data structure of size (6) that for a point $x \in \mathbb{R}^d$ will provide access to $U \cap \pi(x)$ in $O(1)$ time per accessed element. Having that query time is $O(t \cdot \log(Q))$ because, as argued before, for each cube it's enough to consider only that many samples to compute a good estimator. Similarly as in the previous case for repeated queries time drops to $O(1)$.

C.0.2 Ball carving partition

Scheme B: Now let's analyze Ball carving partitions with assumption (9). The situation here is much more complicated and the implementation is much more involved. To compute g we need access to an $\epsilon/4$-net N that covers supp($\mathcal{D}$). We create N on the fly. I.e., we start with $N = \emptyset$ and when a query $q \in \text{supp}(\mathcal{D})$ arrives then if $q \notin \bigcup_{u \in N} B_{\epsilon/4}(u)$ we add q to N. Whenever we add a vertex to N we sample a new permutation σ on N, which corresponds to a new partition. This means that when a point is added to N then g can change. But once the construction process stabilizes then g remains fixed. Using Chebyshev's inequality one can verify that if for $O(1/R(f))$ consecutive queries we don't add new vertices to N then $\bigcup_{u \in N} B_{\epsilon/4}(u)$ contains $1 - O(R(f))$ probability mass of $\mathcal{D}$ with probability $1 - R(f)$. When this event occurs we can stop changing N as the probability mass not covered by N is $O(R(f))$ with high probability. Finally observe that:

$$
\begin{aligned}
|N| &\leq \max_{\substack{N' \subseteq \text{supp}(\mathcal{D}): \\ N' \text{ is } \epsilon/4-net}} |N'| \\
&\leq \min_{\substack{N' \subseteq \text{supp}(\mathcal{D}): \\ N' \text{eq is } \epsilon/8-net}} |N'| =: Q_{max}.
\end{aligned}
$$

Now let's analyze the running time. Consider a query $q \in \mathbb{R}^d$. To compute $g(q)$ we must first check if q should be added to N and this can be done in $O(|N|)$ time. Then we choose a random permutation and locate the set $\pi(q)$ to which q belongs (also in $O(|N|)$ time).

After locating $u \in N$ such that $q \in B_R(u) \setminus \bigcup_{w:\sigma(w)<\sigma(u)} B_R(w) = \pi(q)$ we need to sample points uniformly at random from $\pi(q)$ to compute sample mean to estimate $g(q)$. One way to do that is to use Hit-and-Run sampling. To generate a uniformly random point from $\pi(q)$ we generate a sequence $\{x_i\} \subseteq \pi(q)$ according to the following rule:

- $x_0 = q$,
- to generate x_{i+1} from x_i we first pick a random direction v. We find minimal and maximal values such that $x_i + \theta \cdot v \in \pi(q)$. We pick θ^* uniformly from the interval $[\theta_{\min}, \theta_{\max}]$ and we set $x_{i+1} := x_i + \theta^* \cdot v$.

After generating some number of points we declare the last point as a point drawn from $U(\pi(q))$. The time needed to generate one sample is $k \cdot O(|N|)$, where k is the number of iterations we perform.

To get an algorithm with a theoretical guarantee on the running time for sampling points one can resort to an algorithm from Dyer et al. (1991). That algorithm implicitly, in polynomial in d time, samples a point uniformly at random from a convex body. It is possible to adapt the algorithm to the case of non-convex bodies (as our set $\pi(q)$ is not necessarily convex). We can think that $\pi(q)$ is "close" to being convex as it is defined by a carving process with balls of equal radii. Recall from previous discussion that it's enough to have $O(\log(Q))$ samples per set. So in the end if we use this algorithm then the running time for computing $g(p)$ will be $O\left(\text{poly}(d) \cdot \log(Q) \cdot |N| + t \log(Q)\right) = O\left(\text{poly}(d) \cdot Q_{\max} \log(Q_{\max}) + t \log(Q_{\max})\right)$.

Scheme A: If we use (4) instead of (9) then we first perform a preprocessing step in which we sample a set U of unlabeled samples of size (6) (with Q set to $Q_{\max}$). Then we use a greedy algorithm to find a maximal subset $N \subseteq U$ such that for every $u, w \in N, u \neq w$ we have $||u-w||_2 \geq \epsilon/4$. Using Chebyshev's inequality one can argue that with high probability $\bigcup_{u \in N} B_{\epsilon/4}(u)$ contains $1-O(R(f))$ mass of $\mathcal{D}$. We then perform the ball carving partition using N. Then in time $\widetilde{O}(Q_{\max}^2)$ we create a data structure of size (6) that for a point $u \in N$ will provide access to $U \cap (B_R(u) \setminus \bigcup_{w:\sigma(w)<\sigma(u)} B_R(w))$ in $O(1)$ time per accessed element. Then for a query q we need to first locate $u \in N$ such that $q \in \pi(u)$, which takes $O(Q_{\max})$ time and then we compute sample mean in $O(t \cdot \log(Q_{\max}))$ time. So in the end the running time per query is $O(t \cdot \log(Q_{\max}))$. If there is a repeated query for the same set then we can answer it in $O(Q_{\max})$ time.

The $O(Q_{\max})$ factor in both approaches is far from perfect. However there might be hope to decreasing this factor to $2^{O(dd(\text{supp}(M),\epsilon))}$ using locality sensitive hashing techniques (Gionis et al. (1999)) as in principle we only need to check points in the neighborhood of q to determine $\pi(q)$ and in this neighborhood we have only $2^{O(dd(\text{supp}(M),\epsilon))}$ of them. It might also be possible to reduce the running time further which might be an interesting research direction.

Remark 3. *Assume that the data is supported on a lower dimensional manifold of dimension d' and satisfies the assumptions from Theorem 3. Then robustness guarantees of our algorithms improve automatically with d'. That is we don't need to provide d' as the input to our algorithms.*

D Omitted proofs

D.1 Proofs of Section 3

Lemma 1. *For all separable binary classification tasks and all $\epsilon \in \mathbb{R}_{\geq 0}$ we have that:*

$$AR(\epsilon) = S(2\epsilon).$$

Proof. First we prove that $AR(\epsilon) \leq S(2\epsilon)$. Let E be the minimizer set from the definition of $S(2\epsilon)$. Let $f(x) := -1$ if $d(x, M_- \setminus E) \leq \epsilon$ and $f(x) := +1$ otherwise. Then observe that for all $x \in (M_- \setminus E) \cup (M_+ \setminus E)$ there does not exist an η so that $f(x+\eta) \neq h(x)$. Hence $AR(\epsilon) \leq S(2\epsilon)$.

Now we prove that $AR(\epsilon) \geq S(2\epsilon)$. Let f be a classifier with $AR(f,\epsilon) = r$. That means that there exists $A \subseteq \mathbb{R}^d$ such that

- $\mathbb{P}_X(X \in A) \geq 1 - r$,
- for all $x \in A$ we have $\forall\ \eta \in B_\epsilon\ f(x+\eta) = h(x)$.

This means that $\mathbb{R}^d \setminus A$ is a 2ϵ-separator for that binary task, so in turn $S(2\epsilon) \leq r = AR(f,\epsilon)$. □

D.2 Proofs of Section 4

Fact 2. *$dd((\mathbb{R}^d, \ell_2)) \leq 3d$*

Proof. Let $B_\epsilon(0) \subseteq \mathbb{R}^d$ be a ball of radius ϵ for some $\epsilon > 0$. Let N be an $\epsilon/2$-net of $B_\epsilon(0)$. Notice that all balls in $\{B_{\epsilon/4}(u) : u \in N\}$ are pairwise disjoint and that $\bigcup_{u \in N} B_{\epsilon/4}(u) \subseteq B_{5\epsilon/4}(0)$. Hence $|N| \leq \frac{\text{vol}(B_{5\epsilon/4})}{\text{vol}(B_{\epsilon/4})} = 5^d$. □

Lemma 2. *Let (M,d) be a metric space with ϵ-doubling dimension dd. If all pairwise distances in $N \subseteq M$ are at least r then for any point $x \in M$ and radius $r \leq t \leq \epsilon$ we have $|B_t(x) \cap N| \leq 2^{dd\lceil \log \frac{2t}{r} \rceil}$.*

Proof. As $t \leq \epsilon$ we can use the definition of ϵ-doubling dimension and get that $B_t(x)$ can be covered with 2^{dd} balls of radius $t/2$. Iterating that argument, we conclude that $B_t(x)$ can be covered by $2^{dd\lceil \log \frac{2t}{r} \rceil}$ balls of radius $r/2$. But every such ball can contain at most one point from N so $|B_t(x) \cap N|$ is also upper bounded by $2^{dd\lceil \log \frac{2t}{r} \rceil}$. □

D.3 Proofs of Section 5

Corollary 1. *Let $\Pi \sim \mathcal{P}$ be an $(\epsilon, \beta, \delta)$-padded random partition of a metric space (M,d). Then for every distribution $\mathcal{D}$ we have that:*

$$\mathbb{E}_{\Pi\sim\mathcal{P}}[\mathbb{P}_{X\sim\mathcal{D}}[B_{\epsilon/\beta}(X) \not\subseteq \Pi(X)]] \leq \delta.$$

Proof.

$$\begin{aligned}
&\mathbb{E}_{\Pi\sim\mathcal{P}}[\mathbb{P}_{X\sim\mathcal{D}}[B_{\epsilon/\beta}(X) \not\subseteq \Pi(X)]] \\
&= \mathbb{E}_{\Pi\sim\mathcal{P}}[\mathbb{E}_{X\sim\mathcal{D}}[\mathbb{1}_{\{B_{\epsilon/\beta}(X)\not\subseteq\Pi(X)\}}]] \\
&= \mathbb{E}_{X\sim\mathcal{D}}[\mathbb{E}_{\Pi\sim\mathcal{P}}[\mathbb{1}_{\{B_{\epsilon/\beta}(X)\not\subseteq\Pi(X)\}}]] \\
&= \mathbb{E}_{X\sim\mathcal{D}}\left[\mathbb{P}_{\Pi\sim\mathcal{P}}[B_{\epsilon/\beta}(X) \not\subseteq \Pi(X)]\right] \leq \delta.
\end{aligned}$$

□

Lemma 3. *Let Π be a Cube partition with parameter ϵ. Then for every $\beta > 2\sqrt{d}$ it is $\left(\epsilon, \beta, \frac{O(d^{1.5})}{\beta}\right)$-padded.*

Proof. For all $x \in \mathbb{R}^d$, $\text{diam}(\Pi(x)) = \epsilon$ by construction. Let $A = \left[0, \frac{\epsilon}{\sqrt{d}}\right]^d$. This is the set of *all* points of one fundamental cube. Let $G = \left[\frac{\epsilon}{\beta}, \frac{\epsilon}{\sqrt{d}} - \frac{\epsilon}{\beta}\right]^d$ and note that $d\left(G, \mathbb{R}^d \setminus A\right) = \frac{\epsilon}{\beta}$. G represents the set of all *good* points inside A, in the sense that if we center a sphere of radius ϵ/β at one of those points the whole sphere stays contained inside A. Now observe that

$$\frac{\mathrm{vol}(G)}{\mathrm{vol}\,(A)} = \left(1 - \frac{2\sqrt{d}}{\beta}\right)^d \geq 1 - \frac{2 \cdot d^{1.5}}{\beta}. \quad (14)$$

Let v be the shift that generates the partition π. Consider the set $I(v) := \bigcup_{z \in v + \frac{\epsilon}{\sqrt{d}} \cdot \mathbb{Z}^d} (G + z)$. Using (14), we conclude by noting that for every $x \in \mathbb{R}^d$

$$\mathbb{P}_{\Pi \sim \mathcal{P}}[B_{\frac{\epsilon}{\beta}}(x) \not\subseteq \Pi(x)] \leq \mathbb{P}_{V \sim U(A)}[x \notin I(V)] \leq \frac{2d^{\frac{3}{2}}}{\beta}.$$

□

D.4 Proofs of Section 6

Lemma 5. *Let π be an ϵ-bounded partition. For a given f let $g(x) = sgn(\mathbb{E}_{Z \sim \mathcal{D}}[f(Z) | Z \in \pi(x)])$. Then*

$$R(g) \leq 2S(\epsilon) + 2R(f).$$

Proof. Let us first prove the weaker bound $R(g) \leq 3S(\epsilon) + 2R(f)$. Let E be the minimizer set from the definition of $S(\epsilon)$ and $M_- = h^{-1}(\{-1\}), M_+ = h^{-1}(\{1\})$. Then we know that $d(M_- \setminus E, M_+ \setminus E) \geq \epsilon$ and $\mathbb{P}_{X \sim \mathcal{D}}(X \in E) \leq S(\epsilon)$. Let $Q \subseteq M_- \cup M_+$ be the set of missclassified points of f in $M_- \cup M_+$. Observe that

$$\begin{aligned} R(g) &\leq S(\epsilon) + \sum_{u \in N, \hat{\Pi}(u) \cap M_- \neq \emptyset, g(\hat{\Pi}(u)) = +1} \mu(\hat{\Pi}(u)) \\ &\quad + \sum_{u \in N, \hat{\Pi}(u) \cap M_+ \neq \emptyset, g(\hat{\Pi}(u)) = -1} \mu(\hat{\Pi}(u)) \\ &\leq S(\epsilon) + \sum_{\substack{u \in N, \hat{\Pi}(u) \cap M_- \neq \emptyset, \\ g(\hat{\Pi}(u)) = +1}} 2\mu(\hat{\Pi}(u) \cap (Q \cup E)) \\ &\quad + \sum_{\substack{u \in N, \hat{\Pi}(u) \cap M_+ \neq \emptyset, \\ g(\hat{\Pi}(u)) = -1}} 2\mu(\hat{\Pi}(u) \cap (Q \cup E)) \\ &\leq S(\epsilon) + 2(\mu(Q) + \mu(E)) \\ &\leq 3S(\epsilon) + 2R(f). \end{aligned}$$

To see that the claimed stronger bound is valid note the following. Every point in E will appear either in exactly one of the two sums or it will be counted by the term $S(E)$. In the first two cases it is weighted by a factor 2 and in the second case it is weighted by a factor 1. This gives rise to the term $3S(E)$. But no point of E appears in both of those cases. We can therefore tighten this term to $2S(E)$. □

Lemma 6. *For all $\epsilon > 0$ and any binary classification task with underlying distribution $\mathcal{D}$ if there exists an $(\epsilon\beta, \beta, \delta)$-padded random partition Π of $supp(\mathcal{D})$ then the following conditions hold. There exists a randomized algorithm ALG that given black-box access to classifier f produces a classifier g such that in expectation over the random choices of ALG:*

$$AR(g, \epsilon) \leq 2S(\epsilon\beta) + 2R(f) + \delta$$

and if $AR(\epsilon) > 0$ then:

$$AR(g, \epsilon) \leq \frac{2S(\epsilon\beta)}{S(2\epsilon)} AR(\epsilon) + 2R(f) + \delta.$$

Proof. We will prove that Algorithm 1 invoked with f and $\Pi \sim \mathcal{P}$ satisfies the statement of the Lemma. By Fact 1

$$AR(g, \epsilon) \leq R(g) + \mathbb{P}_{X \sim \mathcal{D}}[g \neg \text{ constant on } B_\epsilon(X)]. \quad (15)$$

By Lemma 5 we have:

$$R(g) \leq 2S(\epsilon\beta) + 2R(f). \quad (16)$$

Moreover, by Corollary 1 we have that:

$$\mathbb{E}_{\Pi \sim \mathcal{P}}[\mathbb{P}_{X \sim \mathcal{D}}[B_\epsilon(X) \not\subseteq \Pi(X)]] \leq \delta. \quad (17)$$

But we also know from the definition of g that

$$\mathbb{P}_{X \sim \mathcal{D}}[g \text{ is not constant on } B_\epsilon(X)] \leq \mathbb{P}_{X \sim \mathcal{D}}[B_\epsilon(X) \not\subseteq \Pi(X)]. \quad (18)$$

Combining (15),(16),(17) and (18) we get that in expectation over the random choices of the algorithm

$$\begin{aligned} AR(g, \epsilon) &\leq 2S(\epsilon\beta) + 2R(f) + \delta \\ &= \frac{2S(\epsilon\beta)}{S(2\epsilon)} AR(\epsilon) + 2R(f) + \delta, \end{aligned}$$

where in the last equality we used Lemma 1. Note that the last inequality is only valid if $AR(\epsilon) > 0$. □

E Oblivious adversary

Let's consider the model where the adversary has full knowledge of the base classifier f and the code of the algorithm ALG that produces g but doesn't have access to random bits used by ALG. Then the following is true:

Theorem 5. *For every separable binary classification task in $\mathbb{R}^d$ and for every $\epsilon \in \mathbb{R}_+$ there exists a randomized algorithm ALG that, given black-box access to $f : \mathbb{R}^d \to \{-1, 1\}$, provides query access to a function $g : \mathbb{R}^d \to \{-1, 1\}$ such that:*

- $R(g) \leq 2S(\epsilon) + 2R(f)$,
- *For every* $x, x' \in \mathbb{R}^d$ *we have that:*

$$\mathbb{P}_{ALG}[g(x) \neq g(x')] \leq O\left(\frac{\|x - x'\|_2 \cdot \sqrt{d}}{\epsilon}\right).$$

Proof. The proof of this theorem is an adaptation of a random partition technique from Charikar et al. (1998). This paper presents an algorithm that creates a random partition that is $(\epsilon, O(\sqrt{d}))-Lipshitz$ (a notion similar to *padded* partitions), that is a random partition that is ϵ-*bounded* and for every $x, x' \in \mathbb{R}^d$:

$$\mathbb{P}[\Pi(x) \neq \Pi(x')] \leq O\left(\frac{\|x - x'\|_2 \cdot \sqrt{d}}{\epsilon}\right).$$

Using this partition ALG creates g using the framework from Algorithm 1. One can verify that this g satisfies the statements of the theorem. □

Remark 4. *We note that the Algorithm from Charikar et al. (1998) is very similar to the random partition from Definition 9 as it also performs a version of ball carving. Based on this similarity, it is tempting to conjecture that the ball carving partition from Definition 9 is* $\left(\epsilon, O(\sqrt{d})\right)-Lipshitz$ *also. We leave this as an interesting open question. Moreover, we note that the Algorithm from Charikar et al. (1998) can be easily adapted to any* ℓ_p *norm achieving* $\left(\epsilon, O(d^{1/2p})\right) - Lipshitz$ *partition for* $1 \leq p \leq 2$ *and* $\left(\epsilon, O(d^{1-1/p})\right) - Lipshitz$ *partition for* $p > 2$. *This means that using this technique one can get adversarial robustness guarantees for any* ℓ_p *norm for* $p \geq 1$.

Now observe that Theorem 5 gives us an algorithm $\mathcal{A}$ that is robust against any oblivious adversary. The algorithm works as follows: for a series of queries $x'_1, x'_2, \dots \in \mathbb{R}^d$ (x'_i's are inputs crafted by the adversary), for every i, $\mathcal{A}$ using ALG from Theorem 5, recomputes a new g_i to answer query x'_i. We know that $R(g_i) \leq 2S(\epsilon) + 2R(f)$ and moreover for every x, x' we have $\mathbb{P}_{ALG}[g_i(x) \neq g_i(x')] \leq O\left(\frac{\|x-x'\|_2 \cdot \sqrt{d}}{\epsilon}\right)$. This means that no matter what the strategy of the adversary is (this strategy might depend on $g_1(x'_1), \dots, g_{i-1}(x'_{i-1})$) the probability that the adversary will be able to construct two points such that $\|x_i - x'_i\|_2 \leq t$ and $g_i(x_i) \neq g_i(x'_i)$ is upper bounded by $O\left(\frac{t \cdot \sqrt{d}}{\epsilon}\right)$.

We summarize: For every i, if $X_i \sim \mathcal{D}$ at the i-th step and the adversary creates X'_i such that $\|X_i - X'_i\|_2 \leq \epsilon$ then for every α:

$$\mathbb{P}_{X_i,\mathcal{A}}(g_i(X'_i) \neq h(X_i)) \leq 2S\left(\frac{\sqrt{d} \cdot \epsilon}{\alpha}\right) + 2R(f) + O(\alpha).$$

Observe the connection to Definition 2 which we restate here for convenience:

$$AR(f, \epsilon) := \mathbb{P}_X(\exists\ \eta \in B_\epsilon\ f(X + \eta) \neq h(X)).$$

The reason that we were able to gain a factor $\sqrt{d}$ in comparison to Theorem 2 is that we didn't need to ensure that a function is constant on a ball $B(x, \epsilon)$. It was enough to show that it is constant for every fixed pair of nearby points as the adversary can only test one point at a time.

This gain comes at a cost as we need to recompute the partition after every query. If one recomputes the partition every k queries then by the union bound the guarantee changes to:

$$\mathbb{P}_{X_i,\mathcal{A}}(g_i(X'_i) \neq h(X_i)) \leq 2S\left(\frac{\sqrt{d} \cdot k \cdot \epsilon}{\alpha}\right) + 2R(f) + O(\alpha).$$