Proceedings of Machine Learning Research vol 75:1–24, 2018 31st Annual Conference on Learning Theory

Active Tolerant Testing

Avrim Blum AVRIM@TTIC.EDU
Toyota Technological Institute at Chicago, Chicago, IL, USA

Lunjia Hu HULJ14@MAILS.TSINGHUA.EDU.CN
Institute for Interdisciplinary Information Sciences, Tsinghua University, Beijing, China

Editors: Sébastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

In this work, we show that for a nontrivial hypothesis class $\mathcal{C}$, we can estimate the distance of a target function f to $\mathcal{C}$ (estimate the error rate of the best $h \in \mathcal{C}$) using substantially fewer labeled examples than would be needed to actually *learn* a good $h \in \mathcal{C}$. Specifically, we show that for the class $\mathcal{C}$ of unions of d intervals on the line, in the active learning setting in which we have access to a pool of unlabeled examples drawn from an arbitrary underlying distribution $\mathcal{D}$, we can estimate the error rate of the best $h \in \mathcal{C}$ to an additive error ϵ with a number of label requests that is *independent of* d and depends only on ϵ. In particular, we make $O(\frac{1}{\epsilon^6}\log\frac{1}{\epsilon})$ label queries to an unlabeled pool of size $O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$. This task of estimating the distance of an unknown f to a given class $\mathcal{C}$ is called *tolerant testing* or *distance estimation* in the testing literature, usually studied in a membership query model and with respect to the uniform distribution. Our work extends that of Balcan et al. (2012) who solved the *non*-tolerant testing problem for this class (distinguishing the zero-error case from the case that the best hypothesis in the class has error greater than ϵ).

We also consider the related problem of estimating the performance of a given learning algorithm $\mathcal{A}$ in this setting. That is, given a large pool of unlabeled examples drawn from distribution $\mathcal{D}$, can we, from only a few label queries, estimate how well $\mathcal{A}$ would perform if the entire dataset were labeled and given as training data to $\mathcal{A}$? We focus on k-Nearest Neighbor style algorithms, and also show how our results can be applied to the problem of hyperparameter tuning (selecting the best value of k for the given learning problem).

Keywords: property testing, agnostic learning, algorithm estimation

1. Introduction

Suppose you are about to embark on a project to label a large quantity of data, such as medical images or street scenes. Your intent is to then feed this data into your favorite learning algorithm for, say, a medical diagnosis or robotic car application. Before embarking on this project, can you, from just a few (perhaps well-chosen) labels, estimate *how well* your algorithm can be expected to perform when trained on the large sample? Here, "few" should mean much less than the number of labeled examples needed for learning, and in particular we will be interested in cases where we can do this with a number of labels that *does not depend on the complexity of the target function*. We consider this question in two related contexts.

Tolerant testing: Here, the goal is to approximate the distance of a target function f to a hypothesis class $\mathcal{C}$. Specifically, consider a hypothesis class $\mathcal{C}$ of VC-dimension d, where d should be thought of as large. (We will generally think of d as large and ϵ as constant.) If we wish to *find* an

ϵ-best hypothesis in $\mathcal{C}$, we will need roughly $O(d/\epsilon^2)$ labeled examples. However, if we just want to estimate what its error rate is without actually finding it, can we do this from less data?

The "realizable" version of this question is the problem of *passive* and *active* property testing, studied by Kearns and Ron (1998) and Balcan et al. (2012). That work considers the problem of distinguishing (a) the case that the target function f belongs to class $\mathcal{C}$ from (b) the case that the target function f is ϵ-far from any concept in $\mathcal{C}$ with respect to the underlying data distribution $\mathcal{D}$. For instance, suppose our data consists of points x on the real line, labeled by f as positive or negative, and we are interested in learning using the class $\mathcal{C}$ consisting of unions of d intervals. This class has VC-dimension $2d$ and so would require $\Omega(d)$ labeled examples to learn. However, Balcan et al. (2012) show that in the active testing framework (one can sample $poly(d)$ *unlabeled* examples for free and then query for the labels of a small number of those examples), one can solve the testing problem using only a constant number of label queries (when ϵ is constant), independent of d.

One limitation of these results, however, is that they do not guarantee to give a meaningful answer when the target function is "almost" in the class $\mathcal{C}$. For instance, suppose f can be perfectly described by a union of 10,000 intervals but is $\epsilon/2$-close to a union of 100 intervals. Then we would like a tester that can say "good enough" at $d = 100$ rather than telling us that we need $d = 10,000$. The tester of Balcan et al. (2012), unfortunately, seems to require f to be $O(\epsilon^3)$-close to a union of d intervals in order to guarantee an output of YES, which is much less than ϵ.

In this work, we give algorithms for such *tolerant testing* (Parnas et al., 2006) for the case of unions of intervals and a few related classes. We can distinguish the case that the best function in $\mathcal{C}$ has error rate $\geq 2\epsilon$ from the case that the best function in $\mathcal{C}$ has error rate $\leq \epsilon$, and more generally we can estimate the error rate α of the best function in the class up to $\pm\epsilon$. Thus, for the first time, from a small number of label queries, we can solve the property-testing analog of the notion of agnostic learning.

One point we wish to make up front: while the classes of functions we consider are fairly simple, such as unions of intervals on the line, we are operating in a challenging model. We would like algorithms that work for any (unknown) underlying data distribution $\mathcal{D}$, not just the uniform distribution, *and* we want algorithms that only query for labels from among examples seen in a poly-sized sample of unlabeled data drawn from $\mathcal{D}$ rather than querying arbitrary points in the input space. These are important conditions for being able to use property testing for machine learning problems.

Algorithm estimation: The second context we consider is that we are given a learning algorithm $\mathcal{A}$ and a large unlabeled sample S of N examples drawn from distribution $\mathcal{D}$. If we were to label all N examples of S and feed them into algorithm $\mathcal{A}$, then $\mathcal{A}$ would produce some hypothesis (call it h_S) with some error rate α. What we would like to do is, by labeling only very few examples in S, and perhaps a few additional examples drawn from $\mathcal{D}$, to estimate the value of α (so that we can determine whether our project of labeling all examples in S is worthwhile).

To get a feel for this problem, one algorithm $\mathcal{A}$ for which this task is easy is 1-Nearest Neighbor (1-NN). This algorithm would produce a hypothesis h_S that on any given query point x predicts the label of the example $x' \in S$ that is nearest to x. For this algorithm, we can easily estimate the error rate of h_S from just a few label queries by repeatedly drawing a random x from $\mathcal{D}$, finding the point $x' \in S$ that is closest to x, and then requesting the labels of x and x' to see if they agree. We only need to repeat this process $O(1/\epsilon^2)$ times in order to estimate the error rate of h_S to $\pm\epsilon$.

This works because h_S is constructed, and makes predictions, in a very local way.[1] In this work, we extend this to different forms of k-Nearest Neighbor algorithms, where the prediction on some point x depends on the k nearest examples in S, developing estimators for which the number of queries *does not depend on* k. This then allows us to use this for *hyperparameter tuning*: determining the (approximately) best value of $k \in \{1, \ldots, N\}$ for the given application.

2. Property Testing Background and Models

2.1. Query Testing (Standard Property Testing)

Given functions f and g over domain X, we define the distance between f and g with respect to distribution $\mathcal{D}$ over X to be

$$\text{dist}_{\mathcal{D}}(f, g) = \Pr_{x \sim \mathcal{D}}[f(x) \neq g(x)]. \tag{1}$$

Given a class $\mathcal{C}$ of functions over domain X and a margin ϵ, a *property tester* distinguishes the case that the input function f is in the class $\mathcal{C}$ from the case that f is ϵ-far from $\mathcal{C}$:

1. if $f \in \mathcal{C}$, the tester accepts f with probability at least $\frac{2}{3}$;
2. if $\forall g \in \mathcal{C}, \text{dist}_{\mathcal{D}}(f, g) > \epsilon$, the tester rejects f with probability at least $\frac{2}{3}$.

Rubinfeld and Sudan (1996) first studied the property testing model assuming X is finite and $\mathcal{D}$ is uniform. We call the testing model of Rubinfeld and Sudan (1996) as *query testing*, because the tester makes queries to access f, i.e., the tester asks for the value of $f(x)$ for some $x \in X$ for each query it makes.

Parnas et al. (2006) first studied the *tolerant* version of property testing: given an additional parameter, the threshold α, to distinguish a function α-close to the class from a function $(\alpha + \epsilon)$-far from the class. In other words,

1. if $\exists g \in \mathcal{C}, \text{dist}_{\mathcal{D}}(f, g) \leq \alpha$, the tester accepts f with probability at least $\frac{2}{3}$;
2. if $\forall g \in \mathcal{C}, \text{dist}_{\mathcal{D}}(f, g) > \alpha + \epsilon$, the tester rejects f with probability at least $\frac{2}{3}$.

They showed tolerant testers for clustering and for monotonicity in the query testing model. Fischer and Fortnow (2005) showed the existence of classes of binary functions that are efficiently query-testable in the non-tolerant case but are not efficiently query-testable in the tolerant case.

Parnas et al. (2006) also considered a similar task called distance approximation: estimating the distance from the function to the class so that with probability at least $\frac{2}{3}$ the output is within $\pm\epsilon$ to the true distance. Note that distance approximation with additive error ϵ implies tolerant testing with margin 2ϵ with the same query complexity. Based on this observation, all the tolerant testers we design in this paper actually perform distance approximation (so we don't need the parameter α) because distance approximation is a slightly more convenient model for our presentation.

1. In contrast, note that estimating the error rate of this algorithm could require a large labeled sample if we only *passively* receive labeled examples. Specifically, suppose the distribution $\mathcal{D}$ is uniform over c clusters and the 1-NN algorithm aims to use $N = c \log \frac{c}{\delta}$ examples, so that with probability at least $1 - \delta$, every cluster has at least one training example in it. We want to distinguish two cases: either every cluster is pure but random so the error rate is roughly 0, or every cluster is 50/50 so the error rate is roughly $\frac{1}{2}$. To distinguish these cases, the estimator needs to see at least two labels in the same cluster, implying an $\Omega(\sqrt{c}) = \Omega(\sqrt{N/\log N})$ passive sample size lower bound.

2.2. Passive Testing (Sample-Based Testing)

Goldreich et al. (1998) first studied testers with the ability to obtain a random sample in addition to making queries so that the tester can potentially work on arbitrary distributions (see Section 2.4 for distribution-free testing), although their algorithmic results remained in the query testing framework over the uniform distribution. Kearns and Ron (1998) developed the first *passive* testers, testers that don't make queries and only rely on the random i.i.d. sample to access the input function f, for a variety of classes with sub-learning sample complexity. Goldreich and Ron (2013) advanced the study of passive testers by providing several general positive results as well as by revealing relations with other testing models.

Proper learning implies testing, simply by testing using the output hypothesis, but passive testing can be substantially harder than *improper* learning. Goldreich et al. (1998) pointed out that the class of k-term-DNF is NP-hard for non-tolerant passive testing while it is efficiently PAC learnable via k-CNF (Pitt and Valiant, 1988), if we require testing and learning on an arbitrary distribution.

The general hardness of *tolerant* passive testing based on hardness of improper *agnostic* learning can be implied from the recent work by Kothari and Livni (2018). They considered the task of refutation: for any fixed distribution $\mathcal{D}$ over domain X, given a sample of example-label pairs $\{(x_i, y_i)\}$ and margin $\epsilon > 0$, to distinguish the following two cases:

1. accept when every (x_i, y_i) is i.i.d. from some distribution $\mathcal{D}'$ over $X \times \{0, 1\}$ with marginal on X being $\mathcal{D}$ and $\exists f \in \mathcal{C}, \Pr_{(x,y)\sim\mathcal{D}'}[f(x) \neq y] \leq \frac{1}{2} - \epsilon$;
2. reject when every x_i is i.i.d. from $\mathcal{D}$ and every y_i is i.i.d. from the uniform distribution over $\{0, 1\}$.

They showed that a refutation algorithm for distribution $\mathcal{D}$ with margin ϵ and sample complexity s implies an improper agnostic learning algorithm for the same distribution with error 3ϵ and sample complexity $O(\frac{s^3}{\epsilon^2})$. We show in Appendix A that the refutation algorithm can be reduced to a tolerant passive tester for arbitrary unknown distributions with threshold $\alpha = \frac{1}{2} - \frac{3\epsilon}{4}$, margin $\frac{\epsilon}{2}$, and sample complexity $\Omega(s)$, implying that tolerant passive testing for arbitrary unknown distributions can't be substantially more sample-efficient than improper agnostic learning for any distribution $\mathcal{D}$ (with some reasonable assumptions about the distribution $\mathcal{D}$).

2.3. Active Testing

Both query testing and passive testing have shortcomings. The assumption of query testing that the tester can make queries to arbitrary points in the domain is usually impractical, while passive testing is too restrictive: for the tolerant case, passive testing can't be substantially more sample-efficient than agnostic learning (recall Section 2.2).

To avoid both shortcomings, Balcan et al. (2012) proposed the active testing model where the tester first receives an unlabeled random i.i.d. sample and then makes queries to points in the sample. While the size of the unlabeled sample might be comparable to the labeled sample complexity for learning, the number of queries the tester makes should be substantially smaller. They showed (non-tolerant) active testers for unions of d intervals and for linear separators.

2.4. Distribution-Free Testing

Distribution-free testing (Goldreich et al., 1998) considers testers that work on arbitrary unknown distributions with the ability to obtain random i.i.d. sample in addition to making queries. Halevy

and Kushilevitz (2003) designed distribution-free testers for low-degree multivariate polynomials, monotone functions, and several other classes.

The difference between distribution-free testing and passive testing (over arbitrary unknown distributions) is that distribution-free testers have the ability to make queries while passive testers don't. However, the query ability is helpful only when we do *non-tolerant* testing where the tester is only required to accept functions in the class, rather than functions having distance 0 to the class with respect to the unknown distribution. For *tolerantly* testing binary functions, we show that distribution-free testing implies passive testing with the same sample complexity (see Section 5 Lemmas 3 and 6) and thus the hardness for tolerant passive testing extends automatically to tolerant distribution-free testing.

3. Our Results and Methods

Tolerant testing: We show (Theorem 8) that in the active testing model, there is a tolerant tester that approximates the distance of a function to the class of unions of d intervals on the line up to an additive error ϵ using $O(\frac{1}{\epsilon^6}\log\frac{1}{\epsilon})$ label queries on $O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$ unlabeled examples.

We begin by assuming data is drawn from the uniform distribution $\mathcal{U}$ over $[0,1]$ and then later generalize to arbitrary distributions $\mathcal{D}$. Our tester evenly partitions $[0,1]$ into $m=\Theta(\epsilon d)$ segments, and focuses on $l = O(\mathrm{poly}(\epsilon^{-1}))$ segments chosen uniformly at random (without replacement). On the union of the chosen segments, we test how close is the function to the class of (roughly) $\frac{dl}{m} = O(\mathrm{poly}(\epsilon^{-1}))$ intervals by a proper agnostic learning algorithm.

One challenge we face is that the number of intervals on each segment might vary drastically, so that the sample of segments is unable to capture the entire information on the whole domain $[0,1]$. To address this challenge, we observe that on each segment, $t=\Theta(\frac{1}{\epsilon^2})$ intervals are sufficient to approximate the distance within an additive error $O(\epsilon)$, and we change the class of unions of d intervals to the class of unions of d intervals *truncated* by t in our algorithm based on this observation. This gives us a (roughly) $(\epsilon, 1+\epsilon)$-bi-criteria tester for the class of unions of intervals, i.e., we estimate the distance up to additive error ϵ and approximate the number of intervals up to a factor of $1+\epsilon$. Balcan et al. (2012) showed that any function that is a union of $(1+\epsilon)d$ intervals has distance $O(\epsilon)$ to a union of d intervals, implying that any function that is α-close to a union of $(1+\epsilon)d$ intervals is $\alpha+O(\epsilon)$ close to a union of d intervals, leading to the uni-criterion tester desired.

Our algorithm implements a reduction that if we can do tolerant testing when d is small $(\mathrm{poly}(\epsilon^{-1}))$, we can do tolerant testing for any d, with query complexity independent of d. We abstract this reduction as the composition lemma (Lemma 7), which may be useful for tolerant testing for other classes. Indeed, the reduction works also for $(\epsilon, 1+\epsilon)$-bi-criteria tolerant testing for surface area for arbitrary $\epsilon > 0$, where the class consists of functions $f : \mathbb{R}^n \to \{0,1\}$ satisfying $f^{-1}(1)$ has a small surface area (see Section 4.2 for related work). For example, consider the class of functions with *Gaussian surface area* (see (Klivans et al., 2008) for definition) at most S with respect to the standard Gaussian distribution over $\mathbb{R}^n$. We can first use $m=\Theta(\epsilon S)$ hyperplanes to evenly partition $\mathbb{R}^n$ into $2m$ parts and focus on $l = O(\mathrm{poly}(\epsilon^{-1}))$ random parts. If we could do tolerant testing on the l parts for $S = O(\mathrm{poly}(\epsilon^{-1}))$, then we could do bi-criteria tolerant testing for general S with query complexity independent of S. However, the difficulty for tolerant testing surface area is that we do not know how to do this even when S is small. Klivans et al. (2008) have shown an agnostic learning algorithm using $n^{O(S^2)}$ samples for the class of concepts with Gaussian surface

area at most S over Gaussian distributions on $\mathbb{R}^n$, but their algorithm is *improper*, not being able to imply a tolerant testing algorithm directly.

To generalize our tester for the class of unions of intervals from the uniform distribution on $[0,1]$ to arbitrary unknown distributions, we show a general relationship between active testing and query testing for arbitrary distributions in Lemmas 2 and 5, which also improves a previous result in (Balcan et al., 2012) by showing that the unlabeled sample complexity of non-tolerant property testing for unions of d intervals on arbitrary unknown distributions can be reduced to $O(\frac{d}{\epsilon}\log\frac{1}{\epsilon})$, from $O(\frac{d^2}{\epsilon^6})$ (implicit) in their original paper. We also generalize the result in (Balcan et al., 2012) for actively testing the class of unions of testable functions to the tolerant case in Appendix B.

Algorithm estimation: For the k-Nearest Neighbor (k-NN) algorithm with soft predictions and pth-power loss (the prediction on a point x is the average label of the k nearest examples to x in a random sample of size N, and we use the pth-power loss to penalize mistakes) we show in Theorem 9 that this loss can be estimated up to an additive error ϵ using $O(\frac{p}{\epsilon^2})$ queries on $N+O(\frac{1}{\epsilon^2})$ unlabeled examples, even when the data distribution is unknown to the estimator. The same result also holds for Weighted Nearest Neighbor algorithms, where the prediction on a point x is a weighted average of the labels of all the examples depending on their distances to x (see Appendix E). For the $O(\frac{p}{\epsilon^2})$ query complexity upper bound, we show a matching lower bound (Theorem 17). In the case where k is a quantity to be optimized, we show an algorithm that finds an approximately-best choice of k up to an additive error ϵ using roughly $O(\frac{p^2\log N}{\epsilon^3})$ queries on roughly $N+O(\frac{p\log N}{\epsilon^3})$ unlabeled examples (Theorem 10). For k-NN with hard predictions (the prediction is a strict majority vote over the k nearest neighbors), we show that it's impossible to estimate the performance with query complexity independent of k (Theorem 18 in Appendix G).

We note that there are three natural but somewhat different ways to model the task of estimating the error rate of algorithm $\mathcal{A}$ trained on dataset S. Let $\mathrm{error}(h_S)$ denote the error rate of hypothesis h_S with respect to distribution $\mathcal{D}$, and let $\hat{\alpha}$ be the output of the estimator $\mathcal{E}$ that estimates $\mathrm{error}(h_S)$. In the first model, we require that $\hat{\alpha}$ be a good estimate of $\mathrm{error}(h_S)$ with probability at least $\frac{2}{3}$ for *any* training set S, even sets S not drawn from $\mathcal{D}$. In the second model, we only require that $\mathcal{E}$ be accurate when S is drawn from $\mathcal{D}$ (that is, the $\frac{2}{3}$ probability is over both the internal randomness in $\mathcal{E}$ and in the draw of S). Finally, in the third model, S is drawn from $\mathcal{D}$ but $\mathcal{E}$ does not have access to it: instead, $\mathcal{E}$ has the ability to draw (a polynomial number of) fresh unlabeled examples and to query points from them. That is,

1. In the first model, we require that $\forall S, \Pr_{\mathcal{E}(S)}[|\hat{\alpha}-\mathrm{error}(h_S)|\le\epsilon]\ge\frac{2}{3}$.
2. In the second model, we require that $\Pr_{S,\mathcal{E}(S)}[|\hat{\alpha}-\mathrm{error}(h_S)|\le\epsilon]\ge\frac{2}{3}$.
3. In the third model, we require that $\Pr_{\mathcal{E}}[|\hat{\alpha}-\mathbb{E}_S[\mathrm{error}(h_S)]|\le\epsilon]\ge\frac{2}{3}$.

Roughly, the first model is the hardest while the third model is the easiest. All our upper bounds and lower bounds in this paper apply to all three models with slight modifications, though for simplicity of presentation we focus on the second model throughout the paper.

4. Additional Related Work

4.1. Testing Unions of d Intervals

We use $\mathcal{I}(d) \subseteq \{0,1\}^{\mathbb{R}}$ to denote the class of functions $f \in \{0,1\}^{\mathbb{R}}$ satisfying that $f^{-1}(1)$ can be written as a union of at most d intervals. Note that for $d \in \mathbb{N}$, the VC-dimension of $\mathcal{I}(d)$ is $2d$.

We use $\mathcal{I}_{\mathcal{D}}(d,\alpha)$ to denote the class of functions that are α-close to $\mathcal{I}(d)$, i.e. $\mathcal{I}_{\mathcal{D}}(d,\alpha) = \{f \in \{0,1\}^{\mathbb{R}} : \exists g \in \mathcal{I}(d), \mathrm{dist}_{\mathcal{D}}(f,g) \leq \alpha\}$. Using this notation, property testing for unions of d intervals is to distinguish $f \in \mathcal{I}(d)$ and $f \notin \mathcal{I}_{\mathcal{D}}(d,\epsilon)$.

In previous work, Kearns and Ron (1998) showed a $(\epsilon, \frac{1}{\epsilon})$-bi-criteria tester for the class of unions of d intervals in the passive testing model, over the uniform distribution $\mathcal{U}$ on $[0,1]$. The tester distinguishes $f \in \mathcal{I}(d)$ and $f \notin \mathcal{I}_{\mathcal{U}}(\frac{d}{\epsilon},\epsilon)$ using $O(\frac{\sqrt{d}}{\epsilon^{1.5}})$ samples. Their tester also works in the standard query testing framework, using $O(\frac{1}{\epsilon})$ queries. Balcan et al. (2012) improved this work by showing that in the active testing framework, there is a uni-criterion testing algorithm that can distinguish $f \in \mathcal{I}(d)$ and $f \notin \mathcal{I}_{\mathcal{U}}(d,\epsilon)$ using $O(\frac{1}{\epsilon^4})$ queries on $O(\frac{\sqrt{d}}{\epsilon^5})$ unlabeled examples. Kothari et al. (2014) slightly improved the query complexity to $O(\frac{1}{\epsilon^{3.5}})$ as the one-dimensional special case when studying the more general problem of testing surface area. Though they were considering the query testing framework, the tester can be easily implemented in the active testing model using the same number of label queries. The tester of Kothari et al. (2014) is similar to that of Balcan et al. (2012), using a "Buffon's Needle"-type algorithm to estimate the "noise sensitivity" of the function being tested. Kothari et al. (2014) provided a stronger analysis than Balcan et al. (2012), roughly allowing the length of the "needle" to be longer by a factor of $\epsilon^{-0.5}$, leading to the improvement of the query complexity. The tester can be generalized to a testing algorithm that distinguishes $f \in \mathcal{I}_{\mathcal{U}}(d,\epsilon_1)$ and $f \notin \mathcal{I}_{\mathcal{U}}(d,\epsilon)$ when $\epsilon_1 = O(\epsilon^{2.5})$ using the same number of queries and unlabeled examples, but can't be directly adapted to the $\epsilon_1 = \frac{\epsilon}{2}$ case, or general tolerant testing. Our tolerant tester for the class of unions of intervals uses a completely different technique.

As pointed out by Balcan et al. (2012), the tester can be generalized from the uniform distribution on $[0,1]$ to any unknown distribution by taking the advantage of unlabeled examples to approximate the CDF of the distribution to enough accuracy using $O(\frac{d^2}{\epsilon^6})$ unlabeled examples. This unlabeled sample complexity is improved to $O(\frac{d}{\epsilon}\log\frac{1}{\epsilon})$ in our paper, by revealing a general relationship between active testing and query testing (see Section 5).

4.2. Testing Surface Area

Kothari et al. (2014) first studied the problem of property testing for the class $\mathcal{C}(A)$ of functions $f : (0,1)^n \to \{0,1\}$ satisfying $f^{-1}(1)$ has surface area at most A over the uniform distribution, which is the high-dimensional version of testing the class of unions of d intervals. They showed a tester that can distinguish $f \in \mathcal{C}(A)$ and f is ϵ-far from $\mathcal{C}(\kappa A)$ with any $\kappa > \frac{4}{\pi} \approx 1.27$ using $O(\frac{1}{\epsilon})$ queries. Neeman (2014) improved the analysis, showing that an essentially identical tester works for any $\kappa > 1$. They both showed similar results for Gaussian surface area.

The testers of Kothari et al. (2014) and Neeman (2014) (together with the tester of Balcan et al. (2012) for unions of intervals) are all based on estimating the "noise sensitivity" of the function and accepting (rejecting) if the "noise sensitivity" is small (large). They showed that a function in the class has a relatively small "noise sensitivity" and a function ϵ-far from the class (with a bi-criteria approximation factor) has a relatively large "noise sensitivity". This argument can't be extended to the *tolerant* case for (ϵ,κ)-bi-criteria testing for arbitrary $\kappa > 1, \epsilon > 0$: while it is

true that a function α-far from the class (with a bi-criteria approximation factor of κ) has a large "noise sensitivity", a function $(\alpha - \epsilon)$-close to the class doesn't necessarily have a smaller "noise sensitivity" (the function has to be $O((\kappa-1)^2\alpha)$-close to the class in order to have a smaller "noise sensitivity" based on the analysis of Neeman (2014)).

5. Relationship between Active Testing and Query Testing

The following Lemma from VC theory shows that when doing non-tolerant testing, the distribution can be assumed to have a finite support with size bounded by a function of the VC-dimension of the concept class.

Lemma 1 *There exists an absolute constant c satisfying the following property. Let $\mathcal{C}$ be a concept class over domain X with VC-dimension d. Let f be any function that is ϵ-far from class $\mathcal{C}$ with respect to distribution $\mathcal{D}$ over X. Let $\mathcal{D}'$ be the uniform distribution over a random iid sample from $\mathcal{D}$ of size at least $\lceil \frac{cd}{\epsilon} \log \frac{1}{\epsilon} \rceil$. Then it holds that f is $\frac{\epsilon}{2}$-far from class $\mathcal{C}$ with respect to distribution $\mathcal{D}'$ with probability at least $\frac{9}{10}$ over the random choice of the sample.*

Therefore, when we perform non-tolerant testing in the active model, we can first sample $\lceil \frac{cd}{\epsilon} \log \frac{1}{\epsilon} \rceil$ unlabeled examples and choose $\mathcal{D}'$ to be the uniform distribution over these examples. The *active* testing task over $\mathcal{D}'$ is almost the same as *query* testing, because the active tester can query arbitrary points in the support of $\mathcal{D}'$, leading to the following Lemma.

Lemma 2 *Let $\mathcal{C}$ be a concept class on ground set X with VC-dimension d. Suppose $\epsilon \in (0, \frac{1}{2})$. Suppose there is a non-tolerant query tester $\mathcal{A}$ with margin $\frac{\epsilon}{2}$ using at most q queries on an* arbitrarily *given distribution with finite support. Suppose all the queries tester $\mathcal{A}$ makes lie in the support of the distribution. Then, there is a non-tolerant active tester $\mathcal{B}$ with margin ϵ using at most $O(q)$ queries on $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$ unlabeled examples for an arbitrary distribution* unknown *to tester $\mathcal{B}$.*

Since Balcan et al. (2012) have an algorithm in the query testing framework that can distinguish $f \in \mathcal{I}(d)$ and $f \notin \mathcal{I}_{\mathcal{D}}(d, \epsilon)$ for arbitrarily given distribution $\mathcal{D}$ using $O(\frac{1}{\epsilon^4})$ queries and the tester only makes queries in the support of $\mathcal{D}$, there is an algorithm in the active testing framework that can distinguish $f \in \mathcal{I}(d)$ and $f \notin \mathcal{I}_{\mathcal{D}}(d, \epsilon)$ using $O(\frac{1}{\epsilon^4})$ queries on $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$ unlabeled examples, even when the distribution $\mathcal{D}$ is unknown, according to Lemma 2. Here, the unlabeled sample complexity is $O(\frac{d}{\epsilon} \log \frac{1}{\epsilon})$, an improvement from $O(\frac{d^2}{\epsilon^6})$ (implicit) in their original paper.

The query tester $\mathcal{A}$ in Lemma 2 is required to query only points in the support of the distribution. This requirement can be removed if $\mathcal{A}$ accepts f when f has distance 0 to $\mathcal{C}$ with respect to the distribution, because in this case the values of f for points outside the support contain no information useful for the tester. The following Lemma shows that such a tester in the distribution-free model implies a passive tester over arbitrary unknown distributions.

Lemma 3 *Suppose we have a non-tolerant distribution-free tester with margin $\frac{\epsilon}{2}$ and sample complexity s for class $\mathcal{C}$ with VC-dimension d, and the tester accepts f when f has distance 0 to the class $\mathcal{C}$. Then there is a non-tolerant passive tester with margin ϵ and sample complexity $O(s)$ for arbitrary unknown distribution $\mathcal{D}$ with no massive points*[2].

2. We say x_0 is a massive point if $\Pr_{x\sim\mathcal{D}}[x = x_0] > 0$.

Proof Imagine we are performing non-tolerant testing in the passive model. The tester first obtains a sample S of size s. When $s < \lceil \frac{cd}{\epsilon} \log \frac{1}{\epsilon} \rceil$, where c is defined in Lemma 1, we enlarge S to a bigger sample S' of size $\lceil \frac{cd}{\epsilon} \log \frac{1}{\epsilon} \rceil$, though only S is revealed to the tester. When $s \geq \lceil \frac{cd}{\epsilon} \log \frac{1}{\epsilon} \rceil$, we simply define $S' = S$. The testing task is then transformed to a testing task over distribution $\mathcal{D}'$ uniform over S' with margin $\frac{\epsilon}{2}$ and success probability at least $(\frac{2}{3})/(\frac{9}{10}) = \frac{20}{27}$, according to Lemma 1. We then perform distribution-free testing with sample S. Note that the distribution $\mathcal{D}$ has no massive points, so with probability 1 no queries made by the distribution-free tester lie in $S' \backslash S$ and thus the queries provide no information useful for the distribution-free tester. Therefore, we can assume the tester gets value 0 for all the queries it makes outside S. ■

Lemmas 1, 2 and 3 can be naturally generalized to the tolerant case as follows.

Lemma 4 *There exists an absolute constant c satisfying the following property. Let $\mathcal{C}$ be a concept class over domain X with VC-dimension d. Let f be any function that has distance α to class $\mathcal{C}$ with respect to distribution $\mathcal{D}$ over X. Let $\mathcal{D}'$ be the uniform distribution over a random iid sample from $\mathcal{D}$ of size at least $\lceil \frac{cd}{\epsilon^2} \log \frac{1}{\epsilon} \rceil$. Then it holds that f has distance within $\alpha \pm \epsilon$ to class $\mathcal{C}$ with respect to distribution $\mathcal{D}'$ with probability at least $\frac{9}{10}$ over the random choice of the sample.*

Lemma 5 *Let $\mathcal{C}$ be a concept class on ground set X with VC-dimension d. Suppose $\epsilon \in (0, \frac{1}{2})$. Suppose there is a tolerant query tester $\mathcal{A}$ with additive error $\frac{\epsilon}{2}$ using at most q queries on an* arbitrarily *given distribution with finite support. Then, there is a tolerant active tester $\mathcal{B}$ with additive error ϵ using at most $O(q)$ queries on $O(\frac{d}{\epsilon^2} \log \frac{1}{\epsilon})$ unlabeled examples for an arbitrary distribution* unknown *to tester $\mathcal{B}$.*

Lemma 6 *Suppose we have a tolerant distribution-free tester with additive error $\frac{\epsilon}{2}$ and sample complexity s for class $\mathcal{C}$ with VC-dimension d. Then there is a tolerant passive tester with additive error ϵ and sample complexity $O(s)$ for arbitrary unknown distribution $\mathcal{D}$ with no massive points.*

6. The Composition Lemma

Balcan et al. (2012) showed that disjoint unions of testable properties are testable in the non-tolerant, active model. We extend their result to tolerant testing in Appendix B. Here, we propose a more general notion of a certain concept class formed by composing smaller concept classes on disjoint ground sets.

Suppose we have m disjoint ground sets $X_1, X_2, \cdots, X_m$ and on each X_i, we have a sequence of concept classes $\mathcal{C}_i^0, \mathcal{C}_i^1, \mathcal{C}_i^2, \cdots \subseteq \{0,1\}^X$. Suppose $\mathcal{C}_i^0 \neq \emptyset$ for all i. We use X to denote the disjoint union $\bigcup_{i=1}^{m} X_i$. For any $d \geq 0$, we define a concept class $\mathcal{P}(d)$ on X to be the class of functions $f \in \{0,1\}^X$ satisfying that $\exists k_1, k_2, \cdots, k_m \in \mathbb{N}$ s.t.

1. $\sum_{i=1}^{m} k_i \leq d$;
2. $\forall 1 \leq i \leq m, f|_{X_i} \in \mathcal{C}_i^{k_i}$.

We call $\mathcal{P}$ a *composition of m additive properties*. Note that $\mathcal{P}(0) = \{f \in \{0,1\}^X : \forall 1 \leq i \leq m, f|_{X_i} \in \mathcal{C}_i^0\}$, matching the definition of a disjoint union of properties in (Balcan et al., 2012). Also note that $\mathcal{P}(0) \neq \emptyset$ because of the assumption that $\mathcal{C}_i^0 \neq \emptyset$ for all i.

For a given $t \geq 0$, we define a composition $\mathcal{P}^t$ in the same way as $\mathcal{P}$ except that we further require every k_i to be at most t, or, $\mathcal{P}^t$ is a composition of m additive properties *truncated by* t.

For any distribution $\mathcal{D}$ over X, we use $\mathcal{P}_{\mathcal{D}}(d, \alpha)$ to denote functions that are α-close to $\mathcal{P}(d)$ with respect to $\mathcal{D}$, i.e. $\mathcal{P}_{\mathcal{D}}(d, \alpha) = \{f \in \{0,1\}^X : \exists g \in \mathcal{P}(d), \mathrm{dist}_{\mathcal{D}}(f, g) \leq \alpha\}$. Similarly, we define $\mathcal{P}^t_{\mathcal{D}}(d, \alpha) = \{f \in \{0,1\}^X : \exists g \in \mathcal{P}^t(d), \mathrm{dist}_{\mathcal{D}}(f, g) \leq \alpha\}$. We say $\mathcal{D}$ is *semi-uniform* if $\forall 1 \leq i \leq m, \Pr_{x \sim \mathcal{D}}[x \in X_i] = \frac{1}{m}$.

An (ϵ, μ)-bi-criteria distance approximation algorithm $\mathrm{Comp}_{\mathcal{D}}(f, (\epsilon, \mu), d)$ for composition $\mathcal{P}$ of additive properties, is an algorithm that takes f, ϵ, μ and d as input and outputs $\hat{\alpha}$ such that $\forall f$

1. $\forall \alpha$ s.t. $f \in \mathcal{P}_{\mathcal{D}}(d, \alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} \leq \alpha + \epsilon$;
2. $\forall \alpha$ s.t. $f \notin \mathcal{P}_{\mathcal{D}}((1+\mu)d, \alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} > \alpha - \epsilon$.

Suppose we have a sequence of indices $1 \leq i_1 < i_2 < \cdots < i_l \leq m$ denoted by $\mathbf{i}$ for short. Let $\mathcal{D}_{\mathbf{i}}$ denote the conditional distribution of $\mathcal{D}$ on $\bigcup_{j=1}^{l} X_{i_j}$. A (d, l, t, ϵ) *distance approximation oracle* is an algorithm taking a length-l sequence $\mathbf{i}$ of indices and $f \in \{0,1\}^X$ as input, and performing $\mathrm{Comp}_{\mathcal{D}_{\mathbf{i}}}(f_{\text{active}}, (\epsilon, 0), d)$ on composition $\mathcal{P}^t$. In other words, this algorithm performs distance approximation on any given l-sub-union (l is typically small) of the m ground sets. For convenience of use, we require the success probability of the oracle to be at least $\frac{11}{12}$. The proof of the following lemma can be found in Appendix C.

Lemma 7 (Composition Lemma) *Suppose $\mathcal{P}$ is the composition of m additive properties defined above. Let $\mathcal{D}$ be a semi-uniform distribution. For parameters $\lambda > 0, \alpha \in [0,1]$ and $\mu, \epsilon \in (0,1)$ taken as input, there exists $l = O(\frac{1}{\epsilon\mu^2} + \frac{1}{\epsilon^2})$ such that we have an algorithm that performs $\mathrm{Comp}_{\mathcal{D}}(f_{\text{active}}, (\epsilon, \mu), \lambda m)$ by calling once a $((1+\frac{\mu}{2})\lambda l, l, \frac{4\lambda}{\epsilon}, \frac{\epsilon}{2})$ distance approximation oracle. Suppose the query complexity and the unlabeled sample complexity of the oracle are q and N, respectively. Then the query complexity and the unlabeled sample complexity of the algorithm are q and $O(\frac{Nm}{l})$, respectively.*

7. Tolerant Testing for Unions of d Intervals

Theorem 8 (main theorem) *Suppose $\mathcal{C}$ is the class of functions $f : \mathbb{R} \to \{0,1\}$ satisfying $f^{-1}(1)$ is a union of at most d intervals for $d > 0$. Given $\epsilon \in (0, \frac{1}{2})$, there is a tolerant tester for $\mathcal{C}$ in the active testing model with respect to an arbitrary unknown distribution $\mathcal{D}$ on $\mathbb{R}$ with additive error ϵ using $O(\frac{1}{\epsilon^6}\log\frac{1}{\epsilon})$ queries on $O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$ unlabeled examples.*

We summarize the proof of Theorem 8 as follows and present the full proof in Appendix D.

Let's first consider the case when $\mathcal{D}$ is the uniform distribution $\mathcal{U}$ over $[0,1]$, and then extend to the arbitrary unknown distribution case. The tester first partitions $[0,1]$ into m pieces, $X_1 = [0, \frac{1}{m}], X_2 = (\frac{1}{m}, \frac{2}{m}], X_3 = (\frac{2}{m}, \frac{3}{m}], \cdots, X_m = (\frac{m-1}{m}, 1]$. $\forall 1 \leq i \leq m, \forall k \in \mathbb{N}$, we define $\mathcal{C}_i^k$ to be the class of binary functions f on X_i such that $f^{-1}(1)$ is a union of at most k intervals. Note that $\mathcal{C}_i^0 \neq \emptyset$. Therefore, we can define $\mathcal{P}$, the composition of m additive properties as in Section 6.

Note that for any $d' > 0$ and any truncation $t > 0$, the concept class $\mathcal{P}^t(d')$ has VC-dimension at most $2d'$. Therefore, according to the VC Theory for agnostic learning, we have a (d', l, t, ϵ') distance approximation oracle using $O(\frac{d'}{\epsilon'^2}\log\frac{1}{\epsilon'})$ queries and unlabeled examples simply by empirical risk minimization. By the Composition Lemma (Lemma 7), the tester calls the oracle once

for $d' = (1+\frac{\mu}{2})(1+\frac{\epsilon}{8})\lambda l, l = O(\frac{1}{\epsilon'\mu} + \frac{1}{\epsilon'^2}), t = \frac{4(1+\frac{\epsilon}{8})\lambda}{\epsilon'}, \epsilon' = \frac{\epsilon}{4}$ and implements an $(\frac{\epsilon}{2}, 1+\mu)$-bi-criteria distance approximation algorithm for $\mathcal{P}((1+\frac{\epsilon}{8})\lambda m) = \mathcal{P}((1+\frac{\epsilon}{8})d)$. We claim that this algorithm is automatically a tolerant tester for the class of unions of d intervals within additive error at most ϵ if we choose $1+\mu = \frac{1+\frac{\epsilon}{4}}{1+\frac{\epsilon}{8}}$.

Note that the active tester for the uniform distribution over $[0,1]$ implies a query tester for the same distribution with the same query complexity. As pointed out by Balcan et al. (2012), the query tester for the uniform distribution over $[0,1]$ then implies a query tester for arbitrary (known) distribution with the same query complexity. According to Lemma 5, the query tester for an arbitrarily given distribution can be finally transformed to an active tester for arbitrary (unknown) distribution with the same query complexity and unlabeled sample complexity $O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$.

8. Estimating the Performance of k-Nearest Neighbor Algorithms

In this section, we develop estimators for estimating the performance of k-Nearest Neighbor (k-NN) algorithms (Fix and Hodges Jr, 1951; Fix and Hodges, 1989; Cover and Hart, 1967).

Let $\mathcal{D}$ be a distribution on a ground set X. Suppose every point $x \in X$ has a (true) label $f(x) \in \{0,1\}$. In addition, we have a distance metric $\mathrm{d} : X \times X \to \mathbb{R}_{\geq 0}$ that is symmetric, nonnegative and satisfies the triangle inequality. The *k-Nearest Neighbor algorithm with soft predictions* (k-NN$^{\text{soft}}$) is given a pool S of unlabeled examples, sampled iid from $\mathcal{D}$, and for any input $x \in X$, finds its k nearest examples $x_1, x_2, \cdots, x_k \in S$ with respect to the distance metric d and outputs $\hat{f}(x) = \frac{1}{k}\sum_{i=1}^{k} f(x_i)$ as an approximation of $f(x)$. In this paper, we assume the k nearest examples are calculated by an oracle M, i.e., when given x and S, M calculates the k nearest examples to x in S. There may be ties when distances to x are compared and we assume M breaks ties according to some (probably random) mechanism.

The *k-Nearest Neighbor algorithm with hard predictions* (k-NN$^{\text{hard}}$) does the same thing as k-NN$^{\text{soft}}$, except that $\hat{f}(x)$ is chosen as the majority vote $I[\frac{1}{k}\sum_{i=1}^{k} f(x_i) > 0.5]$.[3]

For both algorithms, we use $\mathrm{err}_1(x) = |\hat{f}(x) - f(x)|$ to denote the L^1 error on point $x \in X$. For soft prediction, we will penalize the algorithm by taking the pth power of the L^1 error for positive integer p.

8.1. Estimating the Performance of k-NN$^{\text{soft}}$

Given a loss function $\mathrm{loss}(\cdot)$, we can measure the performance of k-NN$^{\text{soft}}$ by its expected loss $\mathbb{E}_x[\mathrm{loss}(\mathrm{err}_1(x))]$. The expectation is over the random draw of x with respect to distribution $\mathcal{D}$ and the randomness of the oracle M when ties occur. In this paper, we focus on the pth-power loss $\mathbb{E}_x[(\mathrm{err}_1(x))^p]$ for positive integer p. Let $\mathcal{E}_{\mathcal{D}}^{\text{soft}}(f, \epsilon, S, k)$ denote the task of estimating the expected loss of a k-NN$^{\text{soft}}$ algorithm up to an additive error ϵ with success probability at least $\frac{2}{3}$. We consider the estimation task in the active model, in which the estimator is only allowed to query labels of examples in an unlabeled pool sampled iid from $\mathcal{D}$. In addition to the given unlabeled pool S from which k-NN$^{\text{soft}}$ would learn, we allow the $\mathcal{E}_{\mathcal{D}}^{\text{soft}}(f_{\text{active}}, \epsilon, S, k)$ estimator to sample fresh unlabeled examples and query their labels. We assume the estimator has access to the oracle M.

3. $I[\cdot]$ is the indicator function of a statement, which takes value 1 if the statement is true and value 0 if the statement is false.

Theorem 9 *Suppose we consider the pth-power loss for $p \in \mathbb{N}^*$. There is an estimator $\mathcal{E}_{\mathcal{D}}^{\text{soft}}(f_{\text{active}}, \epsilon, S, k)$ using $O(\frac{p}{\epsilon^2})$ queries on $N + O(\frac{1}{\epsilon^2})$ unlabeled examples when the unlabeled pool S has size N. The underlying distribution $\mathcal{D}$ is assumed unknown to the estimator. Moreover, the estimator has success probability at least $\frac{2}{3}$ for* any *unlabeled pool S.*

The proof of Theorem 9 is in Appendix E. We will show (Theorem 17 in Appendix G) that the $O(\frac{p}{\epsilon^2})$ query complexity is optimal.

8.2. Finding an Approximately-Best Choice of k

Based on the result in Section 8.1, we are able to construct an algorithm that approximately optimizes the choice of k in the k-NN$^{\text{soft}}$ algorithm.

Suppose we have active access to the true label f with respect to distribution $\mathcal{D}$ over ground set X with distance metric d. Suppose the size of the unlabeled pool S is fixed to be N. We use loss_k to denote the expected loss of the k-NN$^{\text{soft}}$ algorithm and consider how the k-NN$^{\text{soft}}$ algorithm performs with different values of k. We assume the oracle M uses the same tie-breaking mechanism for different values of k. Specifically, given x and S, M arranges the examples in S as $x_1, x_2, \cdots, x_N$ so that $\forall i, \mathrm{d}(x_i, x) \leq \mathrm{d}(x_{i+1}, x)$. $x_1, x_2, \cdots, x_k$ are taken by k-NN$^{\text{soft}}$ as the k nearest neighbors of x for any $k \in \{1, 2, \cdots, N\}$.

We say k is ϵ-approximately-best, if $\forall k' \in \{1, 2, \cdots, N\}, \text{loss}_{k'} \geq \text{loss}_k - \epsilon$. The following theorem states that we can find an ϵ-approximately-best k using a small number of queries. The proof of the theorem is in Appendix F.

Theorem 10 *Suppose k-NN$^{\text{soft}}$ algorithms with an unlabeled pool S of size N are measured by pth-power loss for $p \in \mathbb{N}^*$. Suppose $\epsilon \in (0, \frac{1}{2})$. There is an algorithm that finds an ϵ-approximately-best k w.p. at least $\frac{2}{3}$ using $O(\frac{p^2 \log N}{\epsilon^3}(\log\log N + \log p + \log\frac{1}{\epsilon}))$ queries on $N + O(\frac{p \log N}{\epsilon^3}(\log\log N + \log p + \log\frac{1}{\epsilon}))$ unlabeled examples.*

8.3. Estimating the Performance of k-NN$^{\text{hard}}$

The performance of k-NN$^{\text{hard}}$ is naturally measured by its error rate $\mathbb{E}_x[\text{err}_1(x)]$ and we use $\mathcal{E}_{\mathcal{D}}^{\text{hard}}(f, \epsilon, S, k)$ to denote the corresponding estimation task of estimating the error rate of k-NN$^{\text{hard}}$ up to an additive error ϵ with success probability at least $\frac{2}{3}$.

A trivial estimator achieving this goal using $O(\frac{k}{\epsilon^2})$ queries on $N + O(\frac{1}{\epsilon^2})$ unlabeled examples is to use the empirical mean of $\text{err}_1(x)$ as an estimator of $\mathbb{E}_x[\text{err}_1(x)]$. This estimator is not satisfactory because its query complexity grows with respect to k. In Appendix G, we show (Theorem 18) that this linear growth with respect to k can't be eliminated.

Acknowledgments

This work was supported in part by the National Science Foundation under grants CCF-1525971 and CCF-1800317.

References

Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 21–30. IEEE, 2012.

Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

Eldar Fischer and Lance Fortnow. Tolerant versus intolerant testing for boolean properties. In *Computational Complexity, 2005. Proceedings. Twentieth Annual IEEE Conference on*, pages 135–140. IEEE, 2005.

Evelyn Fix and Joseph Lawson Hodges. Discriminatory analysis. nonparametric discrimination: consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57 (3):238–247, 1989.

Evelyn Fix and Joseph L Hodges Jr. Discriminatory analysis-nonparametric discrimination: consistency properties. Technical report, California Univ Berkeley, 1951.

Oded Goldreich and Dana Ron. On sample-based testers. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 20, pages 4–5, 2013.

Oded Goldreich, Shari Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *Journal of the ACM (JACM)*, 45(4):653–750, 1998.

Shirley Halevy and Eyal Kushilevitz. Distribution-free property testing. In *Approximation, Randomization, and Combinatorial Optimization.. Algorithms and Techniques*, pages 302–317. Springer, 2003.

Emilie Kaufmann, Olivier Cappé, and Aurélien Garivier. On the complexity of best arm identification in multi-armed bandit models. *Journal of Machine Learning Research*, 17(1):1–42, 2016.

Michael Kearns and Dana Ron. Testing problems with sub-learning sample complexity. In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 268–279. ACM, 1998.

Adam R Klivans, Ryan O'Donnell, and Rocco A Servedio. Learning geometric concepts via gaussian surface area. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 541–550. IEEE, 2008.

Pravesh Kothari, Amir Nayyeri, Ryan O'Donnell, and Chenggang Wu. Testing surface area. In *Proceedings of the twenty-fifth annual ACM-SIAM symposium on Discrete algorithms*, pages 1204–1214. SIAM, 2014.

Pravesh K Kothari and Roi Livni. Improper learning by refuting. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 94. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

Joe Neeman. Testing surface area with arbitrary accuracy. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 393–397. ACM, 2014.

Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *Journal of Computer and System Sciences*, 72(6):1012–1042, 2006.

Leonard Pitt and Leslie G Valiant. Computational limitations on learning from examples. *Journal of the ACM (JACM)*, 35(4):965–984, 1988.

Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.

Richard Miles Royall. *A class of non-parametric estimates of a smooth regression function.* PhD thesis, Department of Statistics, Stanford University, 1966.

Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, 1996.

Appendix A. Tolerant Passive Testing Implies Refutation

In this section, we consider a class $\mathcal{C}$ over domain X with VC-dimension d. We are going to build a refutation algorithm (see Section 2.2 for definition) with margin $\epsilon \in (0, \frac{1}{2})$ for a distribution $\mathcal{D}$ by calling a tolerant passive tester over arbitrary unknown distributions with sample complexity s as oracle.

Lemma 11 *There exist universal positive constants c_1, c_2 satisfying the following property. Assume $\mathcal{D}$ satisfies that with probability at least $\frac{11}{12}$ no point appears twice in an s'-sized i.i.d. sample from $\mathcal{D}$ for $s' = \lceil \max\{\frac{c_1 d}{\epsilon^2} \log \frac{1}{\epsilon}, c_2 s\} \rceil$. Suppose there exists a tolerant passive tester $\mathcal{A}$ for $\mathcal{C}$ over arbitrary unknown distribution with threshold $\alpha = \frac{1}{2} - \frac{3\epsilon}{4}$, margin $\frac{\epsilon}{2}$ and sample complexity s. Then there exists a refutation algorithm $\mathcal{B}$ for $\mathcal{C}$ over $\mathcal{D}$ with margin ϵ and sample complexity $c_2 s$.*

Proof Algorithm $\mathcal{B}$ first obtains a $c_2 s$-sized sample S of example label pairs $\{(x_i, y_i)\}$ and declares failure if there exists $y_i \neq y_j$ for $x_i = x_j$. If the algorithm does not declare a failure, it then treat $y_i = f(x_i)$ for some function f and calls $\mathcal{A}$ to distinguish whether f is α-close to $\mathcal{C}$ or f is $(\alpha + \frac{\epsilon}{2})$-far from $\mathcal{C}$ using sample S. Here, the success probability of $\mathcal{A}$ is boosted to at least $\frac{11}{12}$ by repeating c_2 times. $\mathcal{B}$ accepts if $\mathcal{A}$ accepts and $\mathcal{B}$ rejects if $\mathcal{A}$ rejects.

Now, we show the correctness of the algorithm. We first consider the case that every (x_i, y_i) is i.i.d. from a distribution $\mathcal{D}'$ with marginal on X being $\mathcal{D}$ and $\exists g \in \mathcal{C}, \Pr_{(x',y')\sim\mathcal{D}}[g(x') \neq y'] \leq \frac{1}{2} - \epsilon$. We enlarge S to an s'-sized sample S'. Equivalently, we can imagine S' is chosen i.i.d. from $\mathcal{D}'$ first and S is an i.i.d. sample from the uniform distribution $\mathcal{U}$ over S'. With probability at least $\frac{11}{12}$, all the x_i in S' are distinct. For every $x \in X$, we define $\mathrm{error}(x) = \Pr_{(x',y')\sim\mathcal{D}'}[g(x') \neq y' | x' = x]$ and we have $\mathbb{E}_{x\sim\mathcal{D}}[\mathrm{error}(x)] \leq \frac{1}{2} - \epsilon$. According to the Chernoff Bound, with probability at least $\frac{11}{12}$, $\mathbb{E}_{x\sim\mathcal{U}}[\mathrm{error}(x)] \leq \frac{1}{2} - \frac{7\epsilon}{8}$. By the Union Bound, with probability at least $\frac{5}{6}$, all x_i in S' are distinct *and* $\mathbb{E}_{x\sim\mathcal{U}}[\mathrm{error}(x)] \leq \frac{1}{2} - \frac{7\epsilon}{8}$. Conditioned on that, every y_i is independent from others and thus by the Chernoff Bound, with probability at least $\frac{11}{12}$ we have $\Pr_{(x,y)\sim\mathcal{U}}[g(x) \neq y] \leq \frac{1}{2} - \frac{3\epsilon}{4}$. If we unwrap the conditional probability of at least $\frac{5}{6}$, we know with probability at least $\frac{5}{6} \cdot \frac{11}{12}$ that

1. all x_i in S' are distinct;
2. $\Pr_{(x,y)\sim\mathcal{U}}[g(x) \neq y] \leq \alpha$.

Now, if we sample S i.i.d. from $\mathcal{U}$ and feed it to $\mathcal{A}$, we know $\mathcal{A}$ accepts with probability at least $\frac{11}{12}$. Therefore, $\mathcal{B}$ accepts with probability at least $\frac{5}{6} \cdot \frac{11}{12} \cdot \frac{11}{12} \geq \frac{2}{3}$.

Next, we consider the case that every y_i is i.i.d. uniformly chosen from $\{0, 1\}$. Again, we enlarge S to an s'-sized sample S' and imagine S' is chosen i.i.d. from $\mathcal{D}'$ first and S is an i.i.d.

sample from the uniform distribution $\mathcal{U}$ over S'. With probability at least $\frac{11}{12}$, all the x_i in S' are distinct. Conditioned on that, by the Chernoff Bound, for any function g, we have $\Pr_{(x,y)\sim\mathcal{U}}[g(x) \neq y] \geq \frac{1}{2} - \frac{\epsilon}{4} = \alpha + \frac{\epsilon}{2}$ with probability at least $1 - e^{-\epsilon^2 s'/8} = 1 - \epsilon^{c'd/8}$ for some $c' \geq c_1$ satisfying $s' = \frac{c'd}{\epsilon^2}\log\frac{1}{\epsilon}$. By Sauer's Lemma, the number of different g over the chosen x_i in S' is at most $(\frac{es'}{d})^d = (\frac{c'e}{\epsilon^2}\log\frac{1}{\epsilon})^d$. Note that when c_1 is sufficiently large, we always have $\epsilon^{c'd/8} \cdot (\frac{c'e}{\epsilon^2}\log\frac{1}{\epsilon})^d \leq \frac{1}{6}$. Therefore, by the Union Bound, with probability at least $\frac{5}{6}$, we have $\forall g \in \mathcal{C}, \Pr_{(x,y)\sim\mathcal{U}}[g(x) \neq y] \geq \alpha + \frac{\epsilon}{2}$. If we unwrap the conditional probability of at least $\frac{11}{12}$, we know with probability at least $\frac{11}{12} \cdot \frac{5}{6}$ that

1. all x_i in S' are distinct;
2. $\forall g \in \mathcal{C}, \Pr_{(x,y)\sim\mathcal{U}}[g(x) \neq y] \geq \alpha + \frac{\epsilon}{2}$.

Again, if we sample S i.i.d. from $\mathcal{U}$ and feed it to $\mathcal{A}$, we know $\mathcal{A}$ rejects with probability at least $\frac{11}{12}$. Therefore, $\mathcal{B}$ rejects with probability at least $\frac{5}{6} \cdot \frac{11}{12} \cdot \frac{11}{12} \geq \frac{2}{3}$. ■

Appendix B. Distance Approximation for Disjoint Unions of Properties

In this section, we extend the theorem of Balcan et al. (2012) that disjoint unions of testable properties are testable from non-tolerant testing to tolerant testing.

We first introduce the definition of disjoint unions of properties in (Balcan et al., 2012). Suppose the ground set X is partitioned as a disjoint union $\bigcup_{i=1}^{m} X_i$. On every X_i, there is a property (concept class) $\mathcal{C}_i \neq \emptyset$. The disjoint union of these properties is defined to be $\mathcal{C} = \{f \in \{0,1\}^X : \forall 1 \leq i \leq m, f|_{X_i} \in \mathcal{C}_i\}$.

Let $\mathcal{D}$ be a distribution over X. Suppose the conditional distribution of $\mathcal{D}$ on X_i is denoted by $\mathcal{D}_i$ and the probability $\Pr_{x\sim\mathcal{D}}[x \in X_i]$ is denoted by p_i.

Theorem 12 *Suppose $\epsilon \in (0, \frac{1}{2})$. Suppose for every $1 \leq i \leq m$, there is an active tolerant tester $\mathcal{A}$ for $\mathcal{C}_i$ over $\mathcal{D}_i$ with additive error $\frac{\epsilon}{2}$ using at most q queries on N unlabeled examples. Then, there is an active tolerant tester $\mathcal{B}$ for $\mathcal{C}$ over $\mathcal{D}$ with additive error ϵ using at most $O(\frac{q}{\epsilon^2}\log\frac{1}{\epsilon})$ queries on $O(\frac{mN}{\epsilon}\log\frac{1}{\epsilon})$ unlabeled examples. If tester $\mathcal{A}$ can perform on unknown distributions, then tester $\mathcal{B}$ can also perform on unknown distributions, though we need extra $O(\frac{1}{\epsilon^2})$ unlabeled examples.*

Proof Tester $\mathcal{B}$ is constructed as follows. The tester chooses $s = O(\frac{1}{\epsilon^2})$, receives an unlabeled pool of size $O(\frac{mN}{\epsilon}\log s)$ and independently chooses s indices $i_1, i_2, \cdots, i_s$ from $\{1, 2, \cdots, m\}$ according to distribution $\{p_i\}_{1\leq i\leq m}$. This can be achieved by looking at on which X_i's the extra s unlabeled examples are, when the distribution $\mathcal{D}$ is unknown. Then for each $1 \leq j \leq s$, if there are enough $(O(N\log s))$ unlabeled examples lying in X_{i_j}, the tester repeats $\mathcal{A}$ for $O(\log s)$ times to calculate an estimator $\widehat{\text{dist}}_{i_j}$ of the distance from f to $\mathcal{C}$ on $\mathcal{D}_{i_j}$ up to an additive error $\frac{\epsilon}{2}$ with success probability at least $1 - \frac{1}{9s}$;[4] otherwise, define $\widehat{\text{dist}}_{i_j} = 0$. The final output of tester $\mathcal{B}$ is $\frac{1}{s} \cdot \sum_{j=1}^{s} \widehat{\text{dist}}_{i_j}$.

4. Repeat tester $\mathcal{A}$ $O(\log s)$ times and take the median to boost its success probability to at least $1 - \frac{1}{9s}$.

To prove the correctness of the above tester, we first define $\mathrm{dist}_i := \inf_{g\in\mathcal{C}} \mathrm{dist}_{\mathcal{D}_i}(f,g)$ and $\mathrm{dist} := \inf_{g\in\mathcal{C}} \mathrm{dist}_{\mathcal{D}}(f,g)$. Note that $\mathrm{dist} = \sum_{i=1}^{m} p_i \mathrm{dist}_i$.

For every $1 \le i \le m$, we further define $\mathrm{dist}'_i = \begin{cases} \mathrm{dist}_i, & \text{if } p_i \ge \frac{\epsilon}{4m} \\ 0, & \text{if } p_i < \frac{\epsilon}{4m} \end{cases}$ and $\mathrm{dist}''_i = \begin{cases} \mathrm{dist}_i, & \text{if } p_i \ge \frac{\epsilon}{4m} \\ 1, & \text{if } p_i < \frac{\epsilon}{4m} \end{cases}$. Then $\mathrm{dist} - \frac{\epsilon}{4} \le \sum_{i=1}^{m} p_i \mathrm{dist}' \le \sum_{i=1}^{m} p_i \mathrm{dist}'' \le \mathrm{dist} + \frac{\epsilon}{4}$. By the Chernoff Bound, $s = O(\frac{1}{\epsilon^2})$ is enough to make sure with probability at least $1 - \frac{1}{9}$ that $\mathrm{dist} - \frac{\epsilon}{2} < \frac{1}{s}\sum_{j=1}^{s} \mathrm{dist}'_{i_j} \le \frac{1}{s}\sum_{j=1}^{s} \mathrm{dist}''_{i_j} < \mathrm{dist} + \frac{\epsilon}{2}$.

Note that the unlabeled pool has size $O(\frac{mN}{\epsilon}\log s)$, which is enough to make sure that with probability at least $1 - \frac{1}{9}$, for every i_j with $p_{i_j} \ge \frac{\epsilon}{4m}$, there are enough ($O(N \log s)$) unlabeled examples lying in X_{i_j}. Therefore, with probability at least $(1 - \frac{1}{9})(1 - s \cdot \frac{1}{9s}) \ge 1 - \frac{2}{9}$, for all i_j such that $p_{i_j} \ge \frac{p}{4m}$, it holds that $|\widehat{\mathrm{dist}}_{i_j} - \mathrm{dist}_{i_j}| \le \frac{\epsilon}{2}$.

Finally, by the Union Bound, we know with probability at least $1 - \frac{1}{3}$, it holds that $\mathrm{dist} - \epsilon < \frac{1}{s}\sum_{j=1}^{s} \mathrm{dist}'_{i_j} - \frac{\epsilon}{2} \le \frac{1}{s}\sum_{j=1}^{s} \widehat{\mathrm{dist}}_{i_j} \le \frac{1}{s}\sum_{j=1}^{s} \mathrm{dist}''_{i_j} + \frac{\epsilon}{2} < \mathrm{dist} + \epsilon$. ■

Appendix C. Proof of Lemma 7

Before proving Lemma 7, we first show a simple claim about compositions with truncation.

Claim 13 *Suppose the distribution $\mathcal{D}$ is semi-uniform. We have $\mathcal{P}^t_{\mathcal{D}}(d,\alpha) \subseteq \mathcal{P}_{\mathcal{D}}(d,\alpha) \subseteq \mathcal{P}^t_{\mathcal{D}}(d, \alpha + \frac{d}{tm})$.*

Proof $\mathcal{P}^t_{\mathcal{D}}(d,\alpha) \subseteq \mathcal{P}_{\mathcal{D}}(d,\alpha)$ is obvious. To see $\mathcal{P}_{\mathcal{D}}(d,\alpha) \subseteq \mathcal{P}^t_{\mathcal{D}}(d, \alpha + \frac{d}{tm})$, we note that for any $g \in \mathcal{P}(d)$, for each i such that $k_i > t$, substituting a function in $\mathcal{C}_i^0$ for $g|_{X_i}$ causes at most a $\frac{1}{m}$ increase in the distance from $f \in \mathcal{P}_{\mathcal{D}}(d,\alpha)$ to g. An easy observation that $|\{i : k_i > t\}| \le \frac{d}{t}$ given $\sum_{i=1}^{m} k_i \le d$ completes the proof. ■

Proof (of Lemma 7) The algorithm first picks indices $1 \le i_1 < i_2 < \cdots i_l \le m$ uniformly at random for $l = O(\frac{1}{\epsilon\mu^2} + \frac{1}{\epsilon^2})$. Then the algorithm asks for $O(\frac{Nm}{l})$ unlabeled examples to make sure with probability at least $\frac{11}{12}$, there are at least N examples lying in $\bigcup_{j=1}^{l} X_{i_j}$. These examples can be treated as drawn independently at random according to $\mathcal{D}_{\mathbf{i}}$, where $\mathbf{i} = (i_1, i_2, \cdots, i_l)$. Finally, the algorithm calls the oracle to approximate the distance from f to $\mathcal{P}^t((1+\frac{\mu}{2})\lambda l)$ truncated by $t = \frac{4\lambda}{\epsilon}$ on distribution $\mathcal{D}_{\mathbf{i}}$ up to an additive error $\frac{\epsilon}{2}$ using these unlabeled examples and outputs what the oracle outputs.

The correctness of the algorithm follows from the following two lemmas (with proofs in the appendices) and the Union Bound. ■

Lemma 14 *Suppose $t = \frac{4\lambda}{m}$. If $f \in \mathcal{P}_{\mathcal{D}}(\lambda m, \alpha)$, then choosing $l = O(\frac{1}{\epsilon\mu^2} + \frac{1}{\epsilon^2})$ is enough to make sure that with probability at least $\frac{5}{6}$, $f \in \mathcal{P}^t_{\mathcal{D}_{\mathbf{i}}}((1+\frac{\mu}{2})\lambda l, \alpha + \frac{\epsilon}{2})$.*

Lemma 15 *Suppose* $t = \frac{4\lambda}{m}$. *If* $f \notin \mathcal{P}_{\mathcal{D}}((1+\mu)\lambda m, \alpha)$, *then choosing* $l = O(\frac{1}{\epsilon\mu^2} + \frac{1}{\epsilon^2})$ *is enough to make sure that with probability at least* $\frac{5}{6}$, $f \notin \mathcal{P}^t_{\mathcal{D}_\mathbf{i}}((1+\frac{\mu}{2})\lambda l, \alpha - \frac{\epsilon}{2})$.

Proof (of Lemma 14)

By the choice of truncation $t = \frac{4\lambda}{\epsilon}$, according to Claim 13, we know $f \in \mathcal{P}^t_{\mathcal{D}}(\lambda m, \alpha + \frac{\epsilon}{4})$. Suppose $\text{dist}_{\mathcal{D}}(f, g) \leq \alpha + \frac{\epsilon}{4}$ for some $g \in \mathcal{P}^t(\lambda m)$. According to the Multiplicative Chernoff Bound for sampling without replacement, choosing $l = O(\frac{1}{\epsilon\mu^2})$ is enough to make sure that with probability at least $\frac{11}{12}$, $\exists g'$ s.t. $g' \in \mathcal{P}^t((1+\frac{\mu}{2})\lambda l)$ and $\text{dist}_{\mathcal{D}_\mathbf{i}}(g, g') = 0$.[5] According to the Chernoff Bound for sampling without replacement, choosing $l = O(\frac{1}{\epsilon^2})$ is enough to make sure that with probability at least $\frac{11}{12}$, $\text{dist}_{\mathcal{D}_\mathbf{i}}(f, g) \leq \alpha + \frac{\epsilon}{2}$. By the Union Bound, these two events happen at the same time with probability at least $\frac{5}{6}$, and in this case, $f \in \mathcal{P}^t_{\mathcal{D}_\mathbf{i}}((1+\frac{\mu}{2})\lambda l, \alpha + \frac{\epsilon}{2})$. ■

Proof (of Lemma 15) According to Claim 13, we know $f \notin \mathcal{P}^t_{\mathcal{D}}((1+\mu)\lambda m, \alpha)$. Therefore, by definition, there exists $g \in \mathcal{P}^t((1+\mu)\lambda m)$ with the following two properties:[6]

1. $\text{dist}_{\mathcal{D}}(f, g) > \alpha$;
2. $\forall g' \in \mathcal{P}^t((1+\mu)\lambda m), \text{dist}_{\mathcal{D}}(f, g') > \text{dist}_{\mathcal{D}}(f, g) - \frac{\epsilon}{4} \cdot \frac{l}{m}$.

Suppose $g|_{X_i} \in \mathcal{C}_i^{k_i}$ for $k_i \leq t = \frac{4\lambda}{\epsilon}$ satisfying $k := \sum_{i=1}^{m} k_i \leq (1+\mu)\lambda m$. We enlarge k_i to $k_i' \in [k_i, t]$ to make sure that $k' := \sum_{i=1}^{m} k_i' = (1+\mu)\lambda m$.[7] According to the Multiplicative Chernoff Bound for sampling without replacement, choosing $l = O(\frac{1}{\epsilon\mu^2})$ is enough to make sure that with probability at least $\frac{11}{12}$, $\sum_{j=1}^{l} k_{i_j}' \geq (1+\frac{\mu}{2})\lambda l$.

Now suppose it's the case that $\sum_{j=1}^{l} k_{i_j}' \geq (1+\frac{\mu}{2})\lambda l$. Then, according to the second property of g, we know

$$\forall g' \in \mathcal{P}^t((1+\frac{\mu}{2})\lambda l), \text{dist}_{\mathcal{D}_\mathbf{i}}(f, g') > \text{dist}_{\mathcal{D}_\mathbf{i}}(f, g) - \frac{\epsilon}{4}.$$

Otherwise, we can swap g' for g on $\bigcup_{j=1}^{l} X_{i_j}$ causing a violation of the second property of g.

Finally, according to the Chernoff Bound for sampling without replacement, choosing $l = O(\frac{1}{\epsilon^2})$ is enough to make sure that with probability at least $\frac{11}{12}$, $\text{dist}_{\mathcal{D}_\mathbf{i}}(f, g) > \alpha - \frac{\epsilon}{4}$. Therefore, by the Union Bound, with probability at least $\frac{5}{6}$,

$$\forall g' \in \mathcal{P}^t((1+\frac{\mu}{2})\lambda l), \text{dist}_{\mathcal{D}_\mathbf{i}}(f, g') > \text{dist}_{\mathcal{D}_\mathbf{i}}(f, g) - \frac{\epsilon}{4} > \alpha - \frac{\epsilon}{2},$$

a completion of the proof. ■

5. g' is chosen such that $g'|_{X_i} \in \mathcal{C}_i^0$ for all $i \notin \{i_1, i_2, \cdots, i_l\}$ and $g'|_{X_i} = g|_{X_i}$ for all $i \in \{i_1, i_2, \cdots, i_l\}$. The fact that the k_i's of g are bounded between 0 and $t = \frac{4\lambda}{\epsilon}$ allows us to use the Multiplicative Chernoff Bound.
6. E.g., choose g to be the closest or approximately-closest function in the class to f. Note that $\mathcal{P}^t((1+\mu)\lambda m)$ can't be empty, because $\mathcal{P}^t((1+\mu)\lambda m) \supseteq \mathcal{P}^t(0) = \mathcal{P}(0) \neq \emptyset$.
7. k_i' doesn't have to be an integer. Also note that $mt = \frac{4\lambda}{\epsilon} \cdot m > 4\lambda m > (1+\mu)\lambda m$.

Appendix D. Proof of Theorem 8

Proof (of Theorem 8) We use the definitions of $\mathcal{I}(d)$ and $\mathcal{I}_{\mathcal{D}}(d,\alpha)$ in Section 4.1. As pointed out in Section 7, we only need to consider $\mathcal{D}$ as the uniform distribution over $[0,1]$ and we omit it for simplicity.

If $d \leq \frac{8}{\epsilon}$, we can simply do agnostic learning using $O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon}) = O(\frac{1}{\epsilon^3}\log\frac{1}{\epsilon})$ queries and unlabeled examples. So in the rest of the proof, we assume $d > \frac{8}{\epsilon}$. We pick the largest positive integer m satisfying $m \leq \frac{\epsilon d}{8}$ and we define $\lambda = \frac{d}{m} = O(\frac{1}{\epsilon})$.

Since the data distribution is assumed uniform on $[0,1]$, we can assume without loss of generality that our ground set X is $[0,1]$ and $f \in \{0,1\}^X$. We evenly cut X into m pieces: $X_1 = [0,\frac{1}{m}], X_2 = (\frac{1}{m},\frac{2}{m}], X_3 = (\frac{2}{m},\frac{3}{m}], \cdots, X_m = (\frac{m-1}{m},1]$. $\forall 1 \leq i \leq m, \forall k \in \mathbb{N}$, we define $\mathcal{C}_i^k$ to be the class of binary functions f on X_i such that $f^{-1}(1)$ is a union of at most k intervals. Note that $\mathcal{C}_i^0 \neq \emptyset$. Therefore, we can define $\mathcal{P}$, the composition of m additive properties as in Section 6.

Note that for any $d' > 0$ and any truncation $t > 0$, the concept class $\mathcal{P}^t(d')$ has VC-dimension at most $2d'$. Therefore, according to the VC Theory for agnostic learning, for any $\mu \in (0,\frac{1}{2}), \epsilon' = \frac{\epsilon}{4}, l = O(\frac{1}{\epsilon'\mu^2} + \frac{1}{\epsilon'^2})$, we have a $((1+\frac{\mu}{2})(1+\frac{\epsilon}{8})\lambda l, l, \frac{2(1+\frac{\epsilon}{8})\lambda}{\epsilon'}, \epsilon')$ distance approximation oracle using $O(\frac{(1+\frac{\mu}{2})(1+\frac{\epsilon}{8})\lambda l}{\epsilon'^2}\log\frac{1}{\epsilon'}) = O(\frac{l}{\epsilon'^2\epsilon}\log\frac{1}{\epsilon'}) = O((\frac{1}{\epsilon'^3\epsilon\mu^2} + \frac{1}{\epsilon'^4\epsilon})\log\frac{1}{\epsilon'}) = O((\frac{1}{\epsilon^4\mu^2} + \frac{1}{\epsilon^5})\log\frac{1}{\epsilon})$ queries and unlabeled examples simply by empirical risk minimization. By the Composition Lemma (Lemma 7), we have an algorithm that outputs $\hat{\alpha}$ such that $\forall f$,

1. $\forall \alpha$ s.t. $f \in \mathcal{P}((1+\frac{\epsilon}{8})\lambda m, \alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} \leq \alpha + 2\epsilon' (= \alpha + \frac{\epsilon}{2})$;
2. $\forall \alpha$ s.t. $f \notin \mathcal{P}((1+\mu)(1+\frac{\epsilon}{8})\lambda m, \alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} > \alpha - 2\epsilon' (= \alpha - \frac{\epsilon}{2})$.

Choose $1+\mu = \frac{1+\frac{\epsilon}{4}}{1+\frac{\epsilon}{8}}$ and note that $\lambda m = d, \mathcal{I}(d,\alpha) \subseteq \mathcal{P}(d+m,\alpha) \subseteq \mathcal{P}((1+\frac{\epsilon}{8})d,\alpha)$ and $\mathcal{P}((1+\frac{\epsilon}{4})d,\alpha) \subseteq \mathcal{I}((1+\frac{\epsilon}{4})d,\alpha)$, we have $\forall f$,

1. $\forall \alpha$ s.t. $f \in \mathcal{I}(d,\alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} \leq \alpha + \frac{\epsilon}{2}$;
2. $\forall \alpha$ s.t. $f \notin \mathcal{I}((1+\frac{\epsilon}{4})d,\alpha)$, it holds with probability at least $\frac{2}{3}$ that $\hat{\alpha} > \alpha - \frac{\epsilon}{2}$.

This is an $(\frac{\epsilon}{2}, 1+\frac{\epsilon}{4})$-bi-criteria tester for unions of d intervals. According to the Composition Lemma (Lemma 7), the query complexity and the unlabeled sample complexity of the algorithm are $O((\frac{1}{\epsilon^4\mu^2} + \frac{1}{\epsilon^5})\log\frac{1}{\epsilon}) = O(\frac{1}{\epsilon^6}\log\frac{1}{\epsilon})$ and $O((\frac{l}{\epsilon'^2\epsilon}\log\frac{1}{\epsilon'})\cdot\frac{m}{l}) = O(\frac{d}{\epsilon^2}\log\frac{1}{\epsilon})$.

Finally, note that Balcan et al. (2012) revealed a basic property of unions of d intervals that $\mathcal{I}((1+\frac{\epsilon}{4})d) \subseteq \mathcal{I}(d,\frac{\epsilon}{2})$, implying $\mathcal{I}((1+\frac{\epsilon}{4})d,\alpha) \subseteq \mathcal{I}(d,\alpha+\frac{\epsilon}{2})$, which completes the proof. ■

Appendix E. Proof of Theorem 9

Before proving the theorem, we first show a simple estimator that works for any loss function $\mathrm{loss}(\cdot)$ bounded in $[0,1]$ with L-Lipschitz property[8] using $O(\frac{L^2}{\epsilon^4}\cdot\log\frac{1}{\epsilon})$ queries on $N + O(\frac{1}{\epsilon^2})$ unlabeled examples. The estimator runs for $O(\frac{1}{\epsilon^2})$ iterations and in each ith iteration, the estimator samples a

8. We say $\mathrm{loss}(\cdot)$ has L-Lipschitz property if $\forall x_1, x_2 \in [0,1], |\mathrm{loss}(x_1) - \mathrm{loss}(x_2)| \leq L|x_1 - x_2|$.

fresh unlabeled example x and then queries the labels of $w = O(\frac{L^2}{\epsilon^2}\log\frac{1}{\epsilon})$ examples $x_1, x_2, \cdots, x_w$ sampled independently at random uniformly from the k nearest neighbors of x in S. The estimator for this iteration is $E_i = \text{loss}(|\frac{1}{w}\sum_{j=1}^{w} f(x_j) - f(x)|)$. The final output of the estimator is the average of all E_i's for all iterations i.

We prove Theorem 9 by slightly modifying the above estimator's each iteration for pth-power loss. Instead of looking at the labels of w examples, we only need to look at p labels of $x_1, x_2, \cdots, x_p$, still sampled independently at random uniformly from the k nearest neighbors of x in S. In this case, E_i is defined to be $\prod_{j=1}^{p} |f(x_j) - f(x)|$. The final output of the estimator is still the average of E_i's.

Proof (of Theorem 9) We use e_j to denote $|f(x_j) - f(x)|$. To show the above estimator works, we first look at the value we want to estimate: $\mathbb{E}_x[(\text{err}_1(x))^p] = \mathbb{E}_x[|\mathbb{E}_{x_1}[f(x_1) - f(x)]|^p] = \mathbb{E}_x[\mathbb{E}_{x_1}[|f(x_1) - f(x)|]^p] = \mathbb{E}_x[(\mathbb{E}_{x_1}[e_1])^p]$, where x_1 is sampled uniformly from the k nearest neighbors of x in T. Here, we can move the absolute value $|\cdot|$ inside because $f(x_1) - f(x)$ is either always non-negative (when $f(x) = 0$) or always non-positive (when $f(x) = 1$). Note that $x_1, x_2, \cdots, x_p$ are iid, so we know $\mathbb{E}_x[(\text{err}_1(x))^p] = \mathbb{E}_x[(\mathbb{E}_{x_1}[e_1])^p] = \mathbb{E}_x[\mathbb{E}_{x_1,x_2,\cdots,x_p}[e_1 e_2 \cdots e_p]] = \mathbb{E}_{x,x_1,x_2,\cdots,x_p}[\prod_{j=1}^{p} |f(x_j) - f(x)|]$. According to the Chernoff Bound, the empirical mean of $\prod_{j=1}^{p} |f(x_j) - f(x)|$ over $O(\frac{1}{\epsilon^2})$ iid trials approximates the value $\mathbb{E}_x[(\text{err}_1(x))^p]$ within additive error ϵ with probability at least $\frac{2}{3}$, which completes the proof. ■

Theorem 9 also holds naturally for Weighted Nearest Neighbor algorithms (Royall, 1966) with soft predictions, in which $\hat{f}(x)$ is a weighted average of $f(x')$ for all $x' \in S$ where the weights depend on the distances $\text{d}(x', x)$, simply by sampling $x_1, x_2, \cdots, x_p$ iid from S according to the weights.

Appendix F. Proof of Theorem 10

Lemma 16 *Suppose $k_1 \le k_2$ and the loss function $\text{loss}(\cdot)$ is L-Lipschitz. Then, $|\text{loss}_{k_1} - \text{loss}_{k_2}| \le L \cdot (1 - \frac{k_1}{k_2})$.*

Proof When the test point x is chosen, we use $x_1, x_2, \cdots, x_{k_2}$ to denote the closest k_2 points to x in S, arranged in non-decreasing order of their distances to x. Each x_i might be random because ties might be broken randomly. We use e_i to denote $|f(x_i) - f(x)|$. Note that we have

$\text{loss}_{k_1} = \mathbb{E}_{x,x_1,x_2,\cdots,x_{k_1}}[\text{loss}(\frac{1}{k_1}\sum_{i=1}^{k_1} e_i)]$ and $\text{loss}_{k_2} = \mathbb{E}_{x,x_1,x_2,\cdots,x_{k_2}}[\text{loss}(\frac{1}{k_2}\sum_{i=1}^{k_2} e_i)]$. Therefore,

$$\begin{aligned}
&|\text{loss}_{k_1} - \text{loss}_{k_2}| \\
\leq &\mathbb{E}_{x,x_1,x_2,\cdots,x_{k_2}}[|\text{loss}(\frac{1}{k_1}\sum_{i=1}^{k_1} e_i) - \text{loss}(\frac{1}{k_2}\sum_{i=1}^{k_2} e_i)|] \\
\leq &L \cdot \mathbb{E}_{x,x_1,x_2,\cdots,x_{k_2}}[|\frac{1}{k_1}\sum_{i=1}^{k_1} e_i - \frac{1}{k_2}\sum_{i=1}^{k_2} e_i|] \\
= &L \cdot \mathbb{E}_{x,x_1,x_2,\cdots,x_{k_2}}[|(\frac{1}{k_1} - \frac{1}{k_2})\sum_{i=1}^{k_1} e_i - \frac{1}{k_2}\sum_{i=k_1+1}^{k_2} e_i|] \\
\leq &L \cdot \mathbb{E}_{x,x_1,x_2,\cdots,x_{k_2}}[\max\{(\frac{1}{k_1} - \frac{1}{k_2})\sum_{i=1}^{k_1} e_i, \frac{1}{k_2}\sum_{i=k_1+1}^{k_2} e_i\}] \\
\leq &L \cdot \max\{(\frac{1}{k_1} - \frac{1}{k_2}) \cdot k_1, \frac{1}{k_2} \cdot (k_2 - k_1)\} \\
= &L \cdot (1 - \frac{k_1}{k_2})
\end{aligned} \tag{2}$$

■

Proof (of Theorem 10) If we apply Lemma 16 to pth-power loss, which is p-Lipschitz, we know for any $1 \leq \frac{k_2}{k_1} \leq \frac{p}{p-\epsilon}$, it holds that $|\text{loss}_{k_1} - \text{loss}_{k_2}| \leq \epsilon$. If we define $t = \lfloor \log_{\frac{p}{p-\frac{\epsilon}{3}}} N \rfloor, k_{2i} = \lfloor (\frac{p}{p-\frac{\epsilon}{3}})^i \rfloor, k_{2i+1} = \lceil (\frac{p}{p-\frac{\epsilon}{3}})^i \rceil$ for $i = 0, 1, 2, \cdots, t$, then we know $\exists 0 \leq i \leq 2t+1$ such that k_i is $\frac{\epsilon}{3}$-approximately-best. By Theorem 9, we can estimate loss_{k_i} for every $0 \leq i \leq 2t+1$ up to an additive error $\frac{\epsilon}{3}$ using $O(\frac{pt\log t}{\epsilon^2})$ queries on $N + O(\frac{t\log t}{\epsilon^2})$ unlabeled examples.[9] The k_i yielding the smallest approximation of loss_{k_i} is ϵ-approximately-best. Note that $t = O(\frac{p\log N}{\epsilon})$, so the query complexity is $O(\frac{p^2\log N}{\epsilon^3}(\log\log N + \log p + \log\frac{1}{\epsilon}))$ and the unlabeled sample complexity is $N + O(\frac{p\log N}{\epsilon^3}(\log\log N + \log p + \log\frac{1}{\epsilon}))$. ■

Appendix G. Lower Bound Results for Estimating k-Nearest Neighbor Algorithms

Our lower bound results in this section are stronger in the sense that the estimator has query access to f, knows the distribution to be the uniform distribution $\mathcal{U}$ over a finite ground set X and is only supposed to work on some fixed tie-breaking mechanism. Moreover, we don't require the estimator to have success probability at least $\frac{2}{3}$ for *any* S; instead, the success probability is calculated over the random draw of S and the internal randomness of the estimator.

Theorem 17 *Let $\mathcal{U}$ be the uniform distribution over a finite ground set X. There exists a positive constant c such that for any fixed $p \geq 1$, $\epsilon \in (0, \frac{1}{6\sqrt{e}})$ and oracle M using any fixed tie-breaking mechanism, $\mathcal{E}_{\mathcal{U}}^{\text{soft}}(f_{\text{query}}, \epsilon, S, k)$ for pth-power loss requires at least $c \cdot \frac{p}{\epsilon^2}$ queries in the worst case over all finite metric spaces (X, d).*

9. Repeat the estimator $O(\log t)$ times and take the median to boost its success probability to $1 - O(\frac{1}{t})$.

Theorem 18 *There exists a positive constant c such that for any fixed $k \in \mathbb{N}^*, \epsilon \in (0, \frac{1}{4})$ and oracle M using any fixed tie-breaking mechanism, $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ requires at least $c \cdot \frac{k}{\epsilon \log \frac{1}{\epsilon}}$ queries in the worst case.*

Before we prove the above theorems in Sections G.2 and G.3, we first show some related definitions and results in the stochastic multi-armed bandit setting that will be useful in the proofs of the theorems.

G.1. Counting and Approximating the Number of Good Arms

To show query complexity lower bound results for estimating the performance of k-Nearest Neighbor algorithms, we show reductions from two related problems in the stochastic multi-armed bandit setting: counting the number of good arms (CGA) and approximating the number of good arms (AGA).

The setting of stochastic multi-armed bandit problems (Robbins, 1985) is as follows. The algorithm is given n arms, denoted by $\mathbf{A} = (A_1, A_2, \cdots, A_n)$. Each arm is a distribution over $\mathbb{R}$ unknown to the algorithm. The algorithm adaptively accesses these arms to receive values independently sampled according to the distributions.

In this paper, we only consider arms with Bernoulli distributions. When given $\gamma \in (0, \frac{1}{2}]$, we define good arms to be arms with mean at least $\frac{1}{2} + \gamma$ and bad arms to be arms with mean at most $\frac{1}{2} - \gamma$.

The problem of $\text{CGA}(\mathbf{A}, \gamma)$ is, when given $\mathbf{A}$ in which every A_i is either good or bad, to output the number of good arms among the given n arms. The algorithm should output the correct answer with probability at least $\frac{2}{3}$.

The problem of $\text{AGA}(\mathbf{A}, \gamma, \epsilon)$ is a similar task to $\text{CGA}(\mathbf{A}, \gamma)$, except that we only need to approximate the correct answer up to an additive error ϵn.

The following lemma is developed by Kaufmann et al. (2016) as a useful tool for proving lower bounds in the stochastic multi-armed bandit setting.

Lemma 19 (Change of measure) *Suppose $\mathbf{A} = (A_1, A_2, \cdots, A_n)$ and $\mathbf{A}' = (A'_1, A'_2, \cdots, A'_n)$ are two sequences of arms. Suppose algorithm $\mathcal{A}$ takes n arms as input. Suppose $\mathcal{E}$ is an event in the σ-field $\mathcal{F}_T$ for some almost-surely finite stopping time T with respect to the filtration $\{\mathcal{F}_t\}_{t \geq 0}$. Suppose τ_i is the number of queries on A_i made by the algorithm. Then,*

$$\sum_{i=1}^{n} \mathbb{E}_{\mathcal{A},\mathbf{A}}[\tau_i] \text{KL}(A_i, A'_i) \geq D(\Pr_{\mathcal{A},\mathbf{A}}[\mathcal{E}], \Pr_{\mathcal{A},\mathbf{A}'}[\mathcal{E}]).^{10}$$

A simple special case ($n = 1$) of the lemma is that to distinguish a coin with mean μ_1 from a coin with mean μ_2 with success probability at least $1 - \delta$, an algorithm needs at least $\frac{D(1-\delta,\delta)}{D(\mu_1,\mu_2)} = \Omega(\frac{1}{D(\mu_1,\mu_2)} \log \frac{1}{\delta})$ queries in expectation for $\mu_1 \neq \mu_2$ and $0 < \delta \leq \frac{2}{5}$.

10. $\text{KL}(X, Y)$ denotes the Kullback-Leibler divergence from distribution Y to distribution X. If the two distributions X and Y are Bernoulli with means x and y, their Kullback-Leibler divergence is the relative entropy $D(x, y) = x \log \frac{x}{y} + (1 - x) \log \frac{1-x}{1-y}$.

G.2. Proof of Theorem 17

Proof (of Theorem 17) We define $\epsilon' = 6\sqrt{e}\epsilon$. Note that $D(\frac{1-\epsilon'}{2p}, \frac{1}{2p}) = O(\frac{\epsilon'^2}{p})$ for $p \geq 1$ and $\epsilon' \in (0, 1)$. Therefore, we only need to show that a $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ estimator implies an algorithm that distinguishes a coin of mean $\frac{1-\epsilon'}{2p}$ from a coin of mean $\frac{1}{2p}$ with success probability at least $\frac{3}{5}$ using at most the same number of queries. We construct the algorithm in the following way.

The algorithm first constructs a k-NN$^{\text{soft}}$ instance with ground set X and distance metric d. We first choose $k = \lceil \frac{c'p^2}{\epsilon^2} \rceil, b = \lceil \frac{6}{\epsilon} \rceil, N = \lceil c'' \cdot (1+b)k \rceil$ and $m = \lceil \frac{c'''N^2}{1+b} \rceil \geq \frac{6N}{(1+b)\epsilon}$. Here, c', c'' and c''' are sufficiently large constants. X consists of a star with m centers and bm leaves. Each center C has a distance $\mathrm{d}_C \in (1, 2)$ to every leaf in the star and different centers have different values of d_C to avoid ties. The distance between each pair of leaves is 2 and the distance between each pair of centers is 1.

The algorithm then simulates the estimator $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ on this k-NN$^{\text{soft}}$ instance without knowing f beforehand. Every time the estimator queries the label of a new example, it simulates the result as follows. If the example being queried is a leaf, the result is 1. If the example being queried is a center, the result is obtained to be the same result of an independent toss of the coin we want to distinguish. Finally, if the output of $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ is above $\frac{1}{2}[(1-\frac{1}{2p})^p + (1-\frac{1-\epsilon'}{2p})^p]$, the algorithm then guesses the coin to have mean $\frac{1-\epsilon'}{2p}$. Otherwise, the algorithm guesses the coin to have mean $\frac{1}{2p}$.

Now we show that the above algorithm correctly distinguishes the coins with success probability at least $\frac{3}{5}$. The process of the algorithm, by interchanging the randomness of the labels (coin tosses) and the internal randomness of the $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ estimator, can be viewed in the way that the true labels f are determined before we run the $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ estimator. The leaves all have label 1 and each center is independently labeled 0 or 1 according to the result of a toss of the coin. After the labels f are decided, the $\mathcal{E}^{\text{soft}}_{\mathcal{U}}(f_{\text{query}}, \epsilon, S, k)$ estimator is then simulated to approximate the pth-power loss of the k-NN$^{\text{soft}}$ instance up to additive error ϵ with success probability at least $\frac{2}{3}$.

Suppose the coin to be distinguished has mean μ. Note that the total number of points in the ground set is $(1+b)m = \Omega(N^2)$, therefore we can make sure with probability at least $1 - \frac{1}{40}$ that no two unlabeled examples lie on the same point. Because each random example has probability $\frac{1}{1+b}$ to lie in the centers and $N \geq c'' \cdot (1+b)k$, therefore by choosing a sufficiently large c'', we can make sure with probability at least $1 - \frac{1}{40}$ that in the unlabeled sample pool, there are at least k examples lying at the centers. These two events happen at the same time with probability at least $1 - \frac{1}{20}$ by the Union Bound. Conditioned on these two events happening, by a sufficiently large choice of c', among those unlabeled examples lying at the centers, we can make sure that with probability at least $1 - \frac{1}{20}$, the average of the labels of the k examples with smallest d_C is contained in $(\mu - \frac{\epsilon}{6p}, \mu + \frac{\epsilon}{6p})$. All these events happen at the same time with probability at least $(1 - \frac{1}{20})^2 \geq 1 - \frac{1}{10}$, and in this case, every leaf outside the unlabeled pool S has L^1 error in $(1 - \mu - \frac{\epsilon}{6p}, 1 - \mu + \frac{\epsilon}{6p})$ and thus has pth-power loss in $((1-\mu)^p - \frac{\epsilon}{6}, (1-\mu)^p + \frac{\epsilon}{6})$. The total number of leaves in the unlabeled pool S and centers is upper bounded by the size N of the pool plus m, which contributes only a $\frac{N+m}{(b+1)m} \leq \frac{\epsilon}{3}$ fraction of the total number of points. Therefore, with probability at least $1 - \frac{1}{10}$, the average pth-power loss of all points is contained in $((1-\mu)^p - \frac{\epsilon}{2}, (1-\mu)^p + \frac{\epsilon}{2})$.

Note that $(1 - \frac{1-\epsilon'}{2p})^p - (1 - \frac{1}{2p})^p > 3\epsilon$, therefore the algorithm correctly guesses the mean of the coin with probability at least $(1 - \frac{1}{10}) \cdot \frac{2}{3} = \frac{3}{5}$. ■

G.3. Proof of Theorem 18

Lemma 20 *There exists a positive constant c such that for any fixed $k \in \mathbb{N}^*$, $\epsilon \in (0, \frac{1}{4})$ and oracle M using any fixed tie-breaking mechanism, if there is a $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ estimator using at most q queries in the worst case, then there is an $\text{AGA}(\mathbf{A}, \gamma, 2\epsilon)$ algorithm using at most $O(q)$ queries in the worst case where $\gamma = \min\left\{\frac{1}{2}, c \cdot \sqrt{\frac{\log\frac{1}{\epsilon}}{k}}\right\}$.*

The above lemma shows that a query complexity lower bound for $\text{AGA}(\mathbf{A}, \gamma, \epsilon)$ can imply a query complexity lower bound for $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$. $\text{AGA}(\mathbf{A}, \gamma, \epsilon)$ has a simple algorithm requiring $O(\frac{1}{\gamma^2\epsilon^2}\log\frac{1}{\epsilon})$ queries as follows. The algorithm randomly picks $O(\frac{1}{\epsilon^2})$ arms. For each of the picked arms, the algorithm queries it $O(\frac{1}{\gamma^2}\log\frac{1}{\epsilon})$ times and thinks of it as "good" if more than half of the results are positive and "bad" otherwise. The algorithm outputs the fraction of "good" arms among the picked arms.

If we assume the simple $O(\frac{1}{\gamma^2\epsilon^2}\log\frac{1}{\epsilon})$ query complexity for AGA is not improvable, then Lemma 20 implies that the $O(\frac{k}{\epsilon^2})$ query complexity for $\mathcal{E}^{\text{hard}}$ is also not improvable. In other words, if for every sequences $\epsilon_n \to 0$ and $\gamma_n \to 0$, there exists a positive constant c such that $\text{AGA}(\mathbf{A}, \epsilon_i, \gamma_i)$ needs at least $c \cdot \frac{1}{\gamma_i^2\epsilon_i^2}\log\frac{1}{\epsilon_i}$ queries in the worst case, then according to Lemma 20, we know for any sequences $\{k_n\}, \{\epsilon_n\}$ such that $\epsilon_n \to 0$, $\frac{k_n}{\log\frac{1}{\epsilon_n}} \to \infty$, there exists a positive constant c' such that the estimator $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ for k-NN$^{\text{hard}}$ algorithms needs at least $c' \cdot \frac{k_i}{\epsilon_i^2}$ queries in the worst case.

Proof (of Lemma 20) Since the success probability can be boosted by repetition, we only show an $\text{AGA}(\mathbf{A}, \gamma, 2\epsilon)$ algorithm with success probability at least $\frac{3}{5}$. Given any instance of $\text{AGA}(\mathbf{A}, \gamma, 2\epsilon)$ with total number of arms equal to n, the algorithm constructs a ground set X and the distance metric d on it to form a k-NN$^{\text{hard}}$ instance in the following way. We first choose $b = \lceil\frac{3}{\epsilon}\rceil$, $N = \lceil c' \cdot (1+b)n(k + \log\frac{1}{\epsilon})\rceil$ and $m = \lceil\frac{c''N^2}{(1+b)n}\rceil \geq \frac{3N}{(1+b)n\epsilon}$. Here, c' and c'' are sufficiently large constants. X consists of n identical stars, each corresponds to an arm, with the distances between stars to be very large. Each star consists of m centers and bm leaves. Each center C has a distance $\mathrm{d}_C \in (1, 2)$ to every leaf in the same star and different centers have different values of d_C to avoid ties. The distance between each pair of leaves in the same star is 2 and the distance between each pair of centers in the same star is 1.

The algorithm then simulates the estimator $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ on this k-NN$^{\text{hard}}$ instance without knowing f beforehand. Every time the estimator queries the label of a new example, it simulates the result as follows. If the example being queried is a leaf, the result is 0. If the example being queried is a center, the result is obtained to be the same result of an independent query to the corresponding arm. Finally, the algorithm outputs $\hat{\alpha}n$ when the $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ estimator outputs α.

Now we show that the above is an $\text{AGA}(\mathbf{A}, \gamma, 2\epsilon)$ algorithm with success probability at least $\frac{3}{5}$. The process of the algorithm, by interchanging the randomness of the labels (arms) and the internal randomness of the $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ estimator, can be viewed in the way that the true labels f are determined before we run the $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ estimator. The leaves all have labels 0 and each center is independently labeled 0 or 1 according to the result of a query to the corresponding arm. After the labels f are decided, the $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ estimator is then simulated to approximate the error rate of the k-NN$^{\text{hard}}$ instance up to additive error ϵ with success probability at least $\frac{2}{3}$.

Let's say a star is good (bad) if it corresponds to a good (bad) arm. Suppose there are ξn good arms, and thus ξn good stars. Note that there are $(1+b)mn = \Omega(N^2)$ points in the ground set, we can make sure with probability at least $1 - \frac{1}{20}$ that no two unlabeled examples lie on the same point, on which the following discussion is conditioned. Let's first fix a star R whose corresponding arm has mean μ. Because each random example has probability $\frac{1}{(1+b)n}$ to lie in the centers of R and $N \geq c' \cdot (1+b)n(k + \log \frac{1}{\epsilon})$, therefore by choosing a sufficiently large c', we can make sure with probability at most $\frac{\frac{\epsilon}{120}}{1-\frac{1}{20}}$ that in the unlabeled sample pool, there are less than k examples lying at the centers of R. Therefore, by a sufficiently large choice of c, among those unlabeled examples lying at the centers of R, we can make sure that with probability at least $(1 - \frac{\frac{\epsilon}{120}}{1-\frac{1}{20}})(1 - \frac{\epsilon}{200}) \geq 1 - \frac{\frac{\epsilon}{60}}{1-\frac{1}{20}}$, the average of the labels of the k examples with smallest d_C is contained in $(\mu - \gamma, \mu + \gamma)$, or R is *satisfied*. By Markov's Inequality, with probability at least $1 - \frac{\frac{1}{20}}{1-\frac{1}{20}}$, or $1 - \frac{1}{10}$ if we unwrap the conditional probability of $1 - \frac{1}{20}$, at least a $(1 - \frac{\epsilon}{3})$ fraction of all the n stars are satisfied. In a satisfied star, any leaf that is not in the unlabeled pool has L^1 error 1 if the star is good and L^1 error 0 if the star is bad. Note that there are at most N leaves in the unlabeled pool, contributing at most an $\frac{N}{(1+b)mn} \leq \frac{\epsilon}{3}$ fraction of the total number of points. Also there are only mn centers in total, contributing at most an $\frac{mn}{(1+b)mn} \leq \frac{\epsilon}{3}$ fraction of the total number of points. Therefore, with probability at least $1 - \frac{1}{10}$, the average error of all points is contained in $[\xi - \epsilon, \xi + \epsilon]$, which implies that with probability at least $(1 - \frac{1}{10}) \cdot \frac{2}{3} = \frac{3}{5}$, $\hat{\alpha} \in [\xi - 2\epsilon, \xi + 2\epsilon]$. ■

Before proving Theorem 18, we first show a query complexity lower bound for CGA.

Lemma 21 *There exists a universal constant c such that for any fixed $\gamma \in (0, \frac{1}{2}]$ and $n \in \mathbb{N}^*$, $\mathrm{CGA}(\mathbf{A}, \gamma)$ requires at least $c \cdot \frac{n}{\gamma^2}$ queries in the worst case, where n is the number of arms in $\mathbf{A}$.*

Proof (of Lemma 21) We use G to denote the good arm with mean $\frac{1}{2} + \gamma$ and B to denote the bad arm with mean $\frac{1}{2} - \gamma$. Let's first consider the case where each of the n arms is independently chosen to be G or B uniformly at random. Note that we require the probability of success to be at least $\frac{2}{3}$, so $\mathrm{CGA}(\mathbf{A}, \gamma)$ can't always make less than n queries because the probability of success is at most $\frac{1}{2}$ in this case. Therefore, n is an obvious query complexity lower bound and in the rest of the proof we can assume $\gamma < \frac{1}{4}$.

We claim a stronger fact that for any $0 \leq q \leq n$ and any instance consisting of q G's and $n - q$ B's, $\mathrm{CGA}(\mathbf{A}, \gamma)$ needs at least $c \cdot \frac{1}{\gamma^2}$ queries on *every* of the n arms. By symmetry between "good" and "bad", we only show that every G arm needs to be queried at least $c \cdot \frac{1}{\gamma^2}$ times. The reason is as follows. Suppose $\mathbf{A} = (A_1, A_2, \cdots, A_n)$ in which $A_i = G$ for $1 \leq i \leq q$ and $A_i = B$ otherwise. We define $\mathbf{A}' = (A'_1, A'_2, \cdots, A'_n)$ in which $A'_i = G$ for $1 \leq i \leq p-1$ and $A'_i = B$ otherwise. The only difference between $\mathbf{A}$ and $\mathbf{A}'$ is that $A_p = G$ while $A'_p = B$. We use $\mathcal{E}$ to denote the event that $\mathrm{CGA}(\mathbf{A}, \gamma)$ outputs p. By Lemma 19 and $\mathrm{KL}(G, B) = O(\gamma^2)$, we know $\mathbb{E}[\tau_p] \cdot O(\gamma^2) \geq D(\frac{2}{3}, \frac{1}{3}) = \Omega(1)$ and thus $\mathbb{E}[\tau_p] = \Omega(\frac{1}{\gamma^2})$. For similar reasons, we can show for all $1 \leq i \leq p$ that $\mathbb{E}[\tau_i] = \Omega(\frac{1}{\gamma^2})$, which completes the proof. ■

Proof (of Theorem 18) Lemma 21 immediately implies the existence of a positive constant c' such that for any fixed $\epsilon \in (0, \frac{1}{2})$ and $\gamma \in (0, \frac{1}{2}]$, $\mathrm{AGA}(\mathbf{A}, \gamma, \epsilon)$ requires at least $c' \cdot \frac{1}{\gamma^2 \epsilon}$ queries in the

worst case by choosing $n = \lceil \frac{1}{2\epsilon} \rceil - 1$. Then, by Lemma 20, we get an $\Omega(\frac{1}{\left(\min\left\{\frac{1}{2}, \sqrt{\frac{\log \frac{1}{\epsilon}}{k}}\right\}\right)^2} \cdot \frac{1}{2\epsilon}) = \Omega(\frac{k}{\epsilon \log \frac{1}{\epsilon}})$ lower bound for $\mathcal{E}_{\mathcal{U}}^{\text{hard}}(f_{\text{query}}, \epsilon, S, k)$ for $k \in \mathbb{N}^*$ and $\epsilon \in (0, \frac{1}{4})$. ■