Proceedings of Machine Learning Research vol 195:1–23, 2023 36th Annual Conference on Learning Theory

PAC Verification of Statistical Algorithms

Saachi Mutreja SAACHI@BERKELEY.EDU
Jonathan Shafer SHAFERJO@BERKELEY.EDU
University of California, Berkeley

Editors: Gergely Neu and Lorenzo Rosasco

Abstract

Goldwasser et al. (2021) recently proposed the setting of PAC verification, where a hypothesis (machine learning model) that purportedly satisfies the agnostic PAC learning objective is verified using an interactive proof. In this paper we develop this notion further in a number of ways. First, we prove a lower bound of $\Omega\left(\sqrt{d}/\varepsilon^2\right)$ i.i.d. samples for PAC verification of hypothesis classes of VC dimension d. Second, we present a protocol for PAC verification of unions of intervals over $\mathbb{R}$ that improves upon their proposed protocol for that task, and matches our lower bound's dependence on d. Third, we introduce a natural generalization of their definition to verification of general statistical algorithms, which is applicable to a wider variety of settings beyond agnostic PAC learning. Showcasing our proposed definition, our final result is a protocol for the verification of statistical query algorithms that satisfy a combinatorial constraint on their queries.

Keywords: PAC Learning, Interactive Proof Systems, Distribution Testing.

1. Introduction

Comparing what can be computed in a given model of computation versus what can be verified in that model is a recurring theme throughout the fields of computability and computational complexity. The most notorious example is of course the P vs. NP problem, which asks whether the set of decision problems that can be solved in polynomial time equals the set of decision problems whose solution can be verified in polynomial time given a suitable proof string. But the same question has been studied for many other settings and models of computation as well, with prominent examples including L vs. NL (for logspace computation), P vs. $\mathsf{IP} = \mathsf{PSPACE}$ (polytime computation, with an interactive proof) and $\mathsf{MIP}^* = \mathsf{RE}$ (ditto, with multiple quantum provers). The existence of a gap between computing and verifying is sometimes interpreted as capturing the notion of *creativity*, in the sense that finding a solution to a problem might require discovery or inventiveness, while verifying a formal proof for the same is merely rote work.

While this theme has deep roots in the literature and an appealing interpretation, its parallels for *learning* have only recently been explored for the first time. In the context of PAC[1] learning, Goldwasser, Rothblum, Shafer, and Yehudayoff (2021) introduced the setting of *PAC verification*, in which an untrusted prover attempts to convince a verifier that a certain classifier has nearly-optimal loss with respect to a fixed unknown distribution from which the verfier can take random samples.

1. Probably Approximately Correct (PAC) is the standard theoretical model for supervised learning, introduced by Vapnik and Chervonenkis (1968) and Valiant (1984). Agnostic PAC learning is a generalization to the non-realizable case, introduced by Haussler (1992). See also Shalev-Shwartz and Ben-David (2014).

Specifically, they work in the agnostic PAC setting, where the objective is to find a hypothesis h that has nearly-optimal loss in the sense

$$L_{\mathcal{D}}^{0\text{-}1}(h) \leq \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}^{0\text{-}1}(h') + \varepsilon, \tag{1}$$

where $L_{\mathcal{D}}^{0\text{-}1}$ denotes 0-1 population loss and $\mathcal{H}$ is some fixed and known hypothesis class (formal definitions appear in Sections 1.3 and 2.2 below).

Seeing as computational gaps are already well-studied, the main novelty in this setting concerns sample complexity gaps. They show that for some hypothesis classes (but not for others) the number of i.i.d. samples necessary to find a hypothesis with nearly-optimal loss is strictly greater than the number of i.i.d. samples necessary for verifying, with the help of an untrusted prover, that a proposed hypothesis has nearly-optimal loss.

Beyond the (substantial) theoretical motivation, this setting could have meaningful (and timely) real-world applications. First, if a sample complexity gap exists then "verifiable data collection + ML as a service" becomes a viable business model. The provider would collect suitable training data from the desired population distribution, execute a chosen ML algorithm, and subsequently prove to the client that the end result is good with respect to the population distribution. The client would only need a small amount of independent data from the population distribution to determine the veracity of the claim. Beyond this, Goldwasser et al. (2021) envision a variety of other applications, such as more efficient schemes for replicating scientific results in the empirical sciences.

1.1. Our Contributions

PAC verification is novel territory, and very little is currently known. The current paper aims to make some modest steps towards charting this landscape. We focus on studying sample complexity gaps between learning and verifying specifically in terms of the dependence on the VC (Vapnik–Chervonenkis) dimension. We start with showing a lower bound for the sample complexity gap. Prior to our work, one could imagine that some classes would give rise to very large gaps, e.g., $O(\log(d))$ i.i.d. samples for verifying vs. the $\Theta(d)$ samples that are known to be necessary and sufficient for learning, where $d = \mathsf{VC}(\mathcal{H})$. Our first result shows that the gap can be at most quadratic. Namely, for any hypothesis class, PAC verification requires that the verifier use at least $\Omega\left(\sqrt{d}\right)$ i.i.d. random samples.

Second, we show that our lower bound's dependence on the VC dimension is tight in some cases, by improving upon a result of Goldwasser et al. (2021) to obtain a PAC verifier for the class of unions of intervals on $\mathbb{R}$ that uses $O\left(\sqrt{d}\right)$ i.i.d. random samples. The previous result was an upper bound for a weaker notion of verification, that guarantees only that $L_{\mathcal{D}}^{0\text{-}1}(h) \leq 2 \cdot \mathsf{Opt} + \varepsilon$, where $\mathsf{Opt} = \inf_{h' \in \mathcal{H}} L_{\mathcal{D}}^{0\text{-}1}(h')$ (instead of $\mathsf{Opt} + \varepsilon$ as in Eq. (1)). Their result applied only to a specific restriction of the class of unions of intervals, while our technique works for the restricted and for the unrestricted versions of the class.

Third, we take a step towards making the notion of PAC verification more applicable in practical settings. Many ML and data science algorithms that people use in practice, and might like to delegate to an untrusted service, do not obtain (or at least do not provably obtain) the objective of agnostic PAC learning as in Eq. (1). Instead, they obtain some quantity of loss which is typically good enough in practice. With this reality in mind, we introduce a generalization of PAC verification that guarantees that the outcome is competitive with a specific algorithm. Namely, the verifier guarantees that with high probability, the hypothesis h satisfies $L_{\mathcal{D}}^{0\text{-}1}(h) \leq \mathbb{E}\left[L_{\mathcal{D}}^{0\text{-}1}(h_A)\right] + \varepsilon$, where h_A is the (possibly randomized) output of the algorithm (see Theorem 10).

Fourth, we study PAC verification of statistical query algorithms. For a batch $\mathbf{q}$ of statistical queries, we define a notion of *partition size*, denoted $\mathsf{PS}(\mathbf{q})$, which is the number of atoms in the σ-algebra generated by $\mathbf{q}$. We show that whenever this quantity is sufficiently small, there is a sample complexity gap between execution and verification of the statistical query algorithm.

Lastly, we show that there exists a sample complexity gap for a natural example we present, of optimizing a portfolio with advice. Both our lower bound and our upper bound apply to this example.

1.2. Related Works

The study of interactive proofs for properties of distributions was initiated by Chiesa and Gur (2018). They showed general bounds in terms of the support size. However, they did not consider tighter bounds that depend on combinatorial characterizations of the distribution testing property of interest (e.g., bounds that depend on the VC dimension).

The study of PAC verification of a hypothesis class was introduced by Goldwasser, Rothblum, Shafer, and Yehudayoff (2021), who considered interactive proofs for properties of distributions in the specific context of machine learning. In particular, they also considered the relationship between the VC dimension of the class and the sample complexity of verification. They showed a lower bound that is incomparable with our lower bound, and they showed an upper bound for unions of intervals which is weaker than our upper bound. Our definition of PAC verification of an algorithm is closely modeled on their definition.

Recently, there have been a number of works on the general theme of distribution testing and interactive proofs for properties of distributions in the context of machine learning. These include Canetti and Karchmer (2021), Anil, Zhang, Wu, and Grosse (2021), Rubinfeld and Vasilyan (2022) and Herman and Rothblum (2022), among others. Caro, Hinsche, Ioannou, Nietner, and Sweke (2023) studied PAC verification with a quantum prover. Seshia, Sadigh, and Sastry (2022) survey the use of formal methods for verification of AI systems.

1.3. Preliminaries

Notation 1 $\mathbb{N} = \{1, 2, 3, \ldots\}$, *i.e.,* $0 \notin \mathbb{N}$. *For any* $n \in \mathbb{N}$, *we denote* $[n] = \{1, 2, 3, \ldots, n\}$.

Notation 2 *For a set* Ω, *we write* $\Delta(\Omega)$ *to denote the set of all probability measures defined on the measurable space* $(\Omega, \mathcal{F})$, *where* $\mathcal{F}$ *is some fixed* σ*-algebra that is implicitly understood.*

Definition 3 *Let* $\mathcal{P}, \mathcal{Q}$ *be probability measures defined on a measurable space* $(\Omega, \mathcal{F})$. *The total variation distance between* $\mathcal{P}$ *and* $\mathcal{Q}$ *is* $\mathsf{TV}(\mathcal{P}, \mathcal{Q}) = \sup_{A \in \mathcal{F}} |\mathcal{P}(A) - \mathcal{Q}(A)|$.

PAC LEARNING

Definition 4 *Let* $\mathcal{X}$ *be a set, and let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a set of functions. Let* $k \in \mathbb{N}$, $X = \{x_1, x_2, \ldots, x_k\} \subseteq \mathcal{X}$. *We say that* $\mathcal{H}$ *shatters* X *if for any* $y_1, y_2, \ldots, y_k \in \{0,1\}$ *there exists* $h \in \mathcal{H}$ *such that* $h(x_i) = y_i$ *for all* $i \in [k]$. *The Vapnik–Chervonenkis (VC) dimension of* $\mathcal{H}$*,* *denoted* $\mathsf{VC}(\mathcal{H})$*, is the largest* $d \in \mathbb{N}$ *for which there exist a set* $X \subseteq \mathcal{X}$ *of cardinality* d *that is shattered by* $\mathcal{H}$*. If* $\mathcal{H}$ *shatters sets of cardinality arbitrarily large, we say that* $\mathsf{VC}(\mathcal{H}) = \infty$*.*

Throughout most of this paper we use loss functions of the type common in PAC learning, where the loss of a hypothesis with respect to a distribution is defined as the expected loss of that hypothesis on a randomly drawn sample form the distribution, as follows.

Definition 5 *Let* Ω *and* $\mathcal{H}$ *be sets. A* <u>*loss function*</u> *is a function* $L : \Omega \times \mathcal{H} \to [0, 1]$*. Let* $h \in \mathcal{H}$*, and let* $S = (z_1, \ldots, z_m) \in \Omega^m$ *be a vector. The* <u>*empirical loss of* h *with respect to* S</u> *is* $L_S(h) = \frac{1}{m}\sum_{i\in[m]} L(z_i, h)$*. For any distribution* $\mathcal{D} \in \Delta(\Omega)$*, the* <u>*loss of* h *with respect to* $\mathcal{D}$</u> *is* $L_{\mathcal{D}}(h) = \mathbb{E}_{Z\sim\mathcal{D}}[L(Z, h)]$*. The* <u>*loss of* $\mathcal{H}$ *with respect to* $\mathcal{D}$</u> *is* $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h'\in\mathcal{H}} L_{\mathcal{D}}(h')$*.*

The <u>*0-1 loss*</u>*, denoted* $L^{\text{0-1}}$*, is the special case in which* $\mathcal{X}$ *is a set,* $\Omega = \mathcal{X} \times \{0, 1\}$*,* $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$*, and* $L((x, y), h) = \mathbb{1}(h(x) \neq y)$*.*

However, in Theorem 10 below we also consider more general types of loss.

Definition 6 *Let* $\mathcal{X}$ *be a set, and let* $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ *be a class of hypotheses. We say that* $\mathcal{H}$ *is* <u>*agnostically PAC learnable*</u> *if there exist an algorithm* A *and a function* $m_A : [0, 1]^2 \to \mathbb{N}$ *such that for any* $\varepsilon, \delta \in (0, 1)$ *and any distribution* $\mathcal{D} \in \Delta(\mathcal{X} \times \{0, 1\})$*, if* A *receives as input a tuple of* $m_A(\varepsilon, \delta)$ *i.i.d. samples from* $\mathcal{D}$*, then* A *outputs a function* $h \in \mathcal{H}$ *satisfying*

$$\mathbb{P}\left[L^{\text{0-1}}_{\mathcal{D}}(h) \leq L^{\text{0-1}}_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right] \geq 1 - \delta.$$

In words, this means that h *is probably (with confidence* $1 - \delta$*) approximately correct (has loss at most* ε *worse than optimal). The point-wise minimal such function* m_A *is called the* <u>*sample complexity*</u> *of* $\mathcal{H}$*.*

PAC VERIFICATION OF A HYPOTHESIS CLASS

Definition 7 (PAC Verification of a Hypothesis Class; a special case of Goldwasser et al. (2021), Definition 4) *Let* $\mathcal{X}$ *be a set, let* $\mathbb{D} \subseteq \Delta(\mathcal{X} \times \{0, 1\})$ *be a set of distributions, and let* $\mathcal{H} \subseteq \{0, 1\}^{\mathcal{X}}$ *be a class of hypotheses. We say that* $\mathcal{H}$ *is* <u>*PAC verifiable with respect to* $\mathbb{D}$ *using random samples*</u> *if there exist an interactive proof system consisting of a verifier* V *and an honest prover* P *such that for any* $\varepsilon, \delta \in (0, 1)$ *there exist* $m_V, m_P \in \mathbb{N}$ *such that for any* $\mathcal{D} \in \mathbb{D}$*, the following conditions are satisfied:*

- ***Completeness.*** *Let the random variable*

$$h_V = [V(S_V, \varepsilon, \delta), P(S_P, \varepsilon, \delta)] \in \mathcal{H} \cup \{\mathsf{reject}\}$$

denote the output of V *after receiving input* $(S_V, \varepsilon, \delta)$ *and interacting with* P*, which received input* $(S_P, \varepsilon, \delta)$*. Then*

$$\mathbb{P}_{S_V\sim\mathcal{D}^{m_V}, S_P\sim\mathcal{D}^{m_P}}\left[h_V \neq \mathsf{reject} \ \wedge \ \left(L^{\text{0-1}}_{\mathcal{D}}(h_V) \leq L^{\text{0-1}}_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right)\right] \geq 1 - \delta.$$

- ***Soundness.*** *For any (possibly malicious and computationally unbounded) prover* P' *(which may depend on* $\mathcal{D}$*,* ε*, and* δ*), the verifier's output* $h_V = [V(S_V, \varepsilon, \delta), P']$ *satisfies*

$$\mathbb{P}_{S_V\sim\mathcal{D}^{m_V}, S_P\sim\mathcal{D}^{m_P}}\left[h_V = \mathsf{reject} \ \vee \ \left(L^{\text{0-1}}_{\mathcal{D}}(h_V) \leq L^{\text{0-1}}_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right)\right] \geq 1 - \delta.$$

In both conditions, the probability is over the randomness of the samples S_V and S_P, as well as the randomness of V, P and P'.

2. Technical Overview

2.1. Bounds for Verification of VC Classes

Our first result is a lower bound for the number of i.i.d. random samples the verifier requires to successfully PAC verify a class.

Theorem 8 *There exist constants $C, c > 0$ as follows. Let $\varepsilon \in (0,1)$, $\delta = 1/3$, let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a hypothesis class with $\mathsf{VC}(\mathcal{H}) = d \in \mathbb{N}$. Assume that (V, P) is an interactive proof system that PAC verifies $\mathcal{H}$ with parameters (ε, δ) with respect to the set of all distributions $\mathbb{D} = \Delta(\mathcal{X} \times \{0,1\})$, and the verifier V uses $m_V = m_V(d, \varepsilon)$ i.i.d. labeled samples. Then $m_V(d, \varepsilon) \geq (C \cdot \sqrt{d} - c)/\varepsilon^2$.*

Proof Idea This is an application of Le Cam's 'point vs. mixture' method (see Yu, 1997), together with a reduction from distribution testing to PAC verification. Consider distributions where the marginal over the domain is uniform on a fixed $\mathcal{H}$-shattered set of size d. PAC verification requires distinguishing the case of truly random labels (where the loss of the class is $1/2$), from the case where the labels are ε-biased (and the loss of the class is $1/2 - \varepsilon$). An $\Omega\left(\sqrt{d}/\varepsilon^2\right)$ lower bound for distinguishing these two cases is due to Paninski (2008). ■

Our second result shows that the lower bound's dependence on d is tight for a specific class.

Theorem 9 *Let $d \in \mathbb{N}$, and let*

$$\mathcal{H}_d = \left\{ \mathbb{1}_X : \ X = \bigcup_{i \in [d]} [a_i, b_i] \ \wedge \ (\forall i \in [d] : \ 0 \leq a_i \leq b_i \leq 1) \right\} \subseteq \{0,1\}^{[0,1]}$$

be the class of boolean-valued functions over the domain $[0,1]$ that are indicator functions for a union of d intervals. There exists an interactive proof system that PAC verifies the class $\mathcal{H}_d$ with respect to the set of all distributions over $[0,1] \times \{0,1\}$, such that the verifier uses $m_V = O\left(\sqrt{d}\log(1/\delta)\varepsilon^{-2.5}\right)$ random samples, the honest prover uses

$$m_P = O\big((d^2 \log(d/\varepsilon) + \log(1/\delta))\varepsilon^{-4}\big)$$

random samples, and both the verifier and the honest prover run in time polynomial in their numbers of samples.

Proof Idea A discretization of the population distribution is induced by partitioning the domain $[0,1]$ into d/ε intervals, each of which has weight ε/d according to the population distribution. In the discretized distribution, the probability mass from each interval is lumped together into a single arbitrary point in that interval. We show that to find an ε-sub-optimal union of intervals, it suffices to know this discretized distribution. The prover sends the (purported) discretized distribution to the verifier. The verifier uses a distribution identity tester to verify that the provided distribution is a correct discretization of the population distribution. This is possible using $O\left(\sqrt{d}\right)$ samples, because the support of the discretized distribution is of size $O(d)$. ■

2.2. Verification of Statistical Algorithms

Many popular algorithms do not come with provable PAC-like guarantees, but tend to work well in practice. Such heuristics are common in machine learning, data science, optimization, operations research, finance, etc. People might like to delegate the task of collecting data and executing an algorithm on that data to an untrusted party. To capture this notion, our next contribution is a new definition of PAC verification of an *algorithm*.[2] This generalizes the definition of PAC verification of a *hypothesis class* (Theorem 7, introduced by Goldwasser et al., 2021), which corresponds to the special case of PAC verifying an algorithm that is an agnostic PAC learner for the class.

Definition 10 (PAC Verification of an Algorithm) *Let Ω be a set, let $\mathbb{D} \subseteq \Delta(\Omega)$ be a set of distributions, let $\mathcal{H}$ be a set (called the set of* possible outputs*), and for each $\mathcal{D} \in \mathbb{D}$ let $\mathcal{O}_\mathcal{D}$ be an oracle. Let A be a (possibly randomized) algorithm that takes no inputs, has query access to $\mathcal{O}_\mathcal{D}$, and outputs a value $h_A = A^{\mathcal{O}_\mathcal{D}} \in \mathcal{H}$. Let $L: \mathbb{D} \times (\mathcal{H} \cup \{\mathsf{reject}\}) \to [0,1]$ be an arbitrary function*[3]*, let $L_\mathcal{D}(\cdot)$ denote $L(\mathcal{D}, \cdot)$, and let $L_\mathcal{D}(A) = \mathbb{E}[L_\mathcal{D}(h_A)]$, where the expectation is over the randomness of A and of the oracle $\mathcal{O}_\mathcal{D}$. We say that the algorithm A with access to oracles $\{\mathcal{O}_\mathcal{D}\}_{\mathcal{D} \in \mathbb{D}}$ is PAC verifiable with respect to $\mathbb{D}$ by a verification protocol that uses random samples if there exist an interactive proof system consisting of a verifier V and an honest prover P such that for any $\varepsilon, \delta \in (0,1)$ there exist $m_V, m_P \in \mathbb{N}$ such that for any $\mathcal{D} \in \mathbb{D}$, the following conditions are satisfied:*

- ***Completeness.*** *Let the random variable*

$$h_V = [V(S_V, \varepsilon, \delta), P(S_P, \varepsilon, \delta)] \in \mathcal{H} \cup \{\mathsf{reject}\}$$

  denote the output of V after receiving input $(S_V, \varepsilon, \delta)$ and interacting with P, which received input $(S_P, \varepsilon, \delta)$. Then

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}, S_P \sim \mathcal{D}^{m_P}}[h_V \neq \mathsf{reject} \ \wedge \ L_\mathcal{D}(h_V) \leq L_\mathcal{D}(A) + \varepsilon] \geq 1 - \delta.$$

- ***Soundness.*** *For any deterministic or randomized (possibly malicious and computationally unbounded) prover P' (which may depend on $\mathcal{D}$, ε, δ and $\{\mathcal{O}_\mathcal{D}\}_{\mathcal{D} \in \mathbb{D}}$), the verifier's output $h = [V(S_V, \varepsilon, \delta), P']$ satisfies*

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}}[h_V = \mathsf{reject} \ \vee \ L_\mathcal{D}(h_V) \leq L_\mathcal{D}(A) + \varepsilon] \geq 1 - \delta.$$

The probabilities are over the randomness of V, P and P' and of the samples S_V and S_P.

In other words, whereas the definition of Goldwasser et al. (2021) required that the interactive proof system guarantee that a hypothesis is competitive with respect to any hypothesis in $\mathcal{H}$, our definition requires that it be competitive with respect to a specific algorithm.

Remark 11 *PAC verification of an algorithm A requires that $L_\mathcal{D}(h_V) \leq \mathsf{Opt}_A + \varepsilon$ with high probability. Two natural candidate definitions for Opt_A include (1) $\mathsf{Opt}_A = L_\mathcal{D}(h_A)$, and (2) $\mathsf{Opt}_A = \mathbb{E}[L_\mathcal{D}(h_A)]$. Candidate (1) requires that with high probability the verifier's output be at*

2. This notion differs from delegation of computation, in that the data (the input to the algorithm) is collected by the untrusted prover.
3. Note that this is more general than in Theorem 5.

most ε worse than the output of executing algorithm A, while (2) requires that it be at most ε worse than the expected loss of A.

The loss $L_{\mathcal{D}}(h_A)$ is a random variable that depends, inter alia, on the random samples used by A (more generally: on the randomness of the oracle used by A). A crucial aspect of PAC verification is that the verifier use less random samples than are necessary for executing A, and in particular it cannot access the random samples used by A. So the verifier cannot know what loss was obtained in any particular execution of A. Therefore, we reject candidate (1) and adopt candidate (2).

As an application of this new definition, we show that some statistical query algorithms (see Theorems 19 and 21) can be PAC verified via a protocol in which the verifier uses less i.i.d. samples than would be required for simulating the statistical query oracle used by the algorithm. Specifically, for a batch $\mathbf{q}$ of statistical queries, the *partition size* $\mathsf{PS}(\mathbf{q})$ is the number of atoms in the σ-algebra generated by $\mathbf{q}$. If the algorithm uses only batches with small partition size then verification is cheap, as in the following theorem.

Theorem 26 (Informal version) *Let A be a statistical query algorithm that adaptively generates at most b batches of queries with precision τ such that each batch $\mathbf{q}$ satisfies $\mathsf{PS}(\mathbf{q}) \leq s$. Then A is PAC verifiable by an interactive proof system where the verifier uses*

$$m_V = \Theta\left(\frac{\sqrt{s}\log(b/\varepsilon\delta)}{\tau^2} + \frac{\log(1/\varepsilon\delta)}{\varepsilon^2}\right)$$

i.i.d. samples.

Proof Idea The verifier simulates algorithm A. Each time A sends a batch of queries to be evaluated by the statistical query oracle, the verifier sends the queries to the prover, and the prover sends back a vector of purported evaluations. The verifier uses $O\big(\sqrt{s}/\tau^2\big)$ i.i.d. random samples to execute a distribution identity tester (Theorem 15) to verify that the prover's evaluations are correct up to the desired accuracy τ. ■

In particular, Theorem 26 implies the following separation:

Corollary 27 (Informal version) *Let $d \in \mathbb{N}$ and let A be a statistical query algorithm such that each batch of queries generated by A corresponds precisely to a σ-algebra with d atoms. Then simulating A using random samples requires $\Omega\big(d/\tau^2\big)$ random samples, but there exists a PAC verification protocol for A where the verifier uses $O\Big(\sqrt{d}/\tau^2\Big)$ random samples.*

2.3. Examples

Example 1 (Optimizing a portfolio with advice) *Consider a task in which an agent selects a subset S consisting of n items from the set $\Omega = [2n]$. Subsequently, an item $i \in \Omega$ is chosen at random according to a distribution $\mathcal{D} \in \Delta(\Omega)$ that is unknown to the agent, and the agent experiences loss $L(i, S) = \mathbb{1}(i \notin S)$.*

To help make an optimal decision, the agent has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}^m$. Let $\mathcal{H} = \binom{\Omega}{n}$ denote the collection of subsets of size n that the agent could select. $\mathsf{VC}(\mathcal{H}) = n$, and therefore estimating the expected loss $L_{\mathcal{D}}(S)$ of each possible choice $S \in \mathcal{H}$ up to precision $\varepsilon > 0$ requires $m_A = \Omega\big((n + \log(1/\delta))/\varepsilon^2\big)$ samples.

By Theorem 27, if the agent can receive advice from an untrusted prover, it can make an ε-optimal choice using $m_V = O\left(\sqrt{n}\log(1/\delta\varepsilon)/\varepsilon^2\right)$ i.i.d. samples. Note that $m_V \ll m_A$ for large n. Furthermore, our expression for m_V is tight in the sense that, by Theorem 8, $\Omega(\sqrt{n})$ samples are necessary for verifying the advice of an untrusted prover. ■

Note that the above example is an instance of verification in our generalized setting (Theorem 10), but it is technically not an instance of PAC verification as previously defined by Goldwasser et al. (2021), e.g., because the distribution has no labels. More generally, Theorem 10 includes verification of *distribution learning*, as follows.

Example 2 (Verification of distribution learning) *Let $\Omega = [n]$. Consider a task in which an agent has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}^m$ from some distribution $\mathcal{D} \in \Delta(\Omega)$ that is unknown to the agent. The agent selects a distribution $\hat{\mathcal{D}} \in \Delta(\Omega)$, and experience loss $L_{\mathcal{D}}\left(\hat{\mathcal{D}}\right) = \mathsf{TV}\left(\hat{\mathcal{D}}, \mathcal{D}\right)$.*

It is well known that to achieve loss at most ε with probability at least $1-\delta$, it is necessary and sufficient to take $m_A = \Theta\left((n + \log(1/\delta))/\varepsilon^2\right)$ samples (Canonne, 2020b, Theorem 1). In contrast, if the agent has access to advice from an untrusted prover then $m_V = O\left(\sqrt{n}\log(1/\delta)\varepsilon^{-2}\right)$ i.i.d. samples are sufficient. The honest prover simply sends the verifier a description of a distribution $\tilde{\mathcal{D}} \in \Delta(\Omega)$ that has loss at most $\varepsilon/\sqrt{n}$. The verifier uses distribution testing (Theorem 15) to decide whether $L_{\mathcal{D}}\left(\tilde{\mathcal{D}}\right) \leq \varepsilon/\sqrt{n}$ or $L_{\mathcal{D}}\left(\tilde{\mathcal{D}}\right) \geq \varepsilon$, and accepts if and only if the former case holds. ■

A large collection of concrete tasks that might be of interest and that fall within the setting of Theorem 10 involve solving various problems on graphs given random samples that convey information about the graph, as follows.

Example 3 (Verification in graphs) *Fix $n \in \mathbb{N}$. For any graph $G = (V, E)$ with $V = [n]$, let $\mathcal{D}_G$ be the uniform distribution on E. The agent does not know G, but it knows n and it has access to an i.i.d. sample $Z = (z_1, \ldots, z_m) \sim \mathcal{D}_G^m$. Consider some standard tasks, such as:*

- *Maximum matching. The agent selects a subset $M \subseteq \binom{V}{2}$ and experiences loss*

$$L_{\mathcal{D}_G}(M) = \min_{M' \in \mathcal{M}} \frac{|M \Delta M'|}{n},$$

where $\mathcal{M}$ is the set of all matchings in G of maximal size.

- *Coloring. The agent selects a function $f: V \to \mathbb{N}$ and experiences loss*

$$L_{\mathcal{D}_G}(f) = \min_{f' \in \mathcal{F}} \frac{\sum_{v \in V} \mathbb{1}\left(f(v) \neq f'(v)\right)}{n}$$

where $\mathcal{F}$ is the set of all valid colorings of G that use a minimal number of colors.

For these tasks, there is an easy lower bound of $m = \Omega(n)$ on the number of samples the agent needs to guarantee loss at most ε with probability at least $1-\delta$ for $\varepsilon = \delta = 0.1$. To see this, consider the family of graphs that consist of a disjoint union of triplets (sets of three vertices), such that each triplet contains a single edge. Because the agent does not know in advance where the edge is in each triplet, finding an approximately maximum matching and an approximate 2-coloring require seeing nearly all the edges in the graph.

However, if we assume that G has maximum degree bounded by a constant (as in the lower bound), then $\mathcal{D}_G$ is a uniform distribution with support size $O(n)$. Hence, given access to advice from an untrusted prover, the agent can solve these tasks using $O(\sqrt{n})$ samples using the verification procedure of Example 2.

To see that $\Omega(\sqrt{n})$ samples are necessary for verification with the help of a prover, consider a family of graphs consisting of a disjoint union of triplets as above, but where only half the triplets contain an edge. Distinguishing between this family and the previous family requires observing a collision (receiving a sample that contains the same edge twice), which requires $\Omega(\sqrt{n})$ samples by the 'birthday paradox'. ■

So far, all our examples involved a quadratic gap between learning and verifying. However, larger gaps are possible if we make strong assumptions on the unknown distribution. One example of this, pointed out by Goldwasser et al. (2021), is that the gap between learning and verifying for *realizable* PAC learning is unbounded. Unbounded gaps can exist also for other tasks as well, as in the following example.

Example 4 (Unbounded gap in a graph task) *Let n, $G = (V, E)$, and $\mathcal{D}_G$ be as in Example 3. Consider the maximal matching tasks under the assumption that E is a perfect matching. Again, there is an easy lower bound of $\Omega(n)$ random samples to guarantee loss at most ε with probability at least $1 - \delta$ for $\varepsilon = \delta = 0.1$ without the help of a prover. To see this, consider a graph that is a disjoint union of sets of four vertices, where each such set contains two disjoint edges. Finding a perfect matching requires seeing an edge from each set.*

In contrast, $m_V = O(\log(1/\delta)/\varepsilon)$ samples are sufficient given advice from an untrusted prover. The protocol is as follows. The prover sends $\tilde{E}$, which purportedly equals E. If $\tilde{E}$ is not a perfect matching then the verifier rejects. Then, the verifier takes m_V samples from $\mathcal{D}_G$, and accepts if and only if all the edges in the sample appear in $\tilde{E}$. For completeness, if $\tilde{E} = E$ then the verifier always accepts. For soundness, if $\left(|\tilde{E}\Delta E|\right)/n \geq \varepsilon$, then $\mathcal{D}_G$ has weight $\Omega(\varepsilon)$ on edges that are not in $\tilde{E}$, and so taking m_V samples is sufficient to ensure that the verifier rejects with probability at least $1 - \delta$. ■

For the maximum matching task, we have seen that under the assumption that G has maximum degree bounded by a constant the sample complexity gap is quadratic, but that the gap is unbounded under the stronger assumption that G is a perfect matching. We view this as a demonstration of the richness of this setting.

3. A Lower Bound for PAC Verification of VC Classes

Theorem 8 is proved via a reduction from the following distribution testing lower bound.

Theorem 12 (Reformulation of Theorem 4 in Paninski, 2008) *Let $d, t \in \mathbb{N}$ and let $\varepsilon \in (0, 1)$. For every $\sigma \in \Sigma = \{\pm 1\}^d$, let $\mathcal{D}_{\sigma,\varepsilon} \in \Delta([2d])$ be a distribution such that for all $i \in [d]$,*

$$\mathcal{D}_{\sigma,\varepsilon}(2i-1) = \frac{1 + \sigma_i \cdot \varepsilon}{2d}, \quad \textit{and} \quad \mathcal{D}_{\sigma,\varepsilon}(2i) = \frac{1 - \sigma_i \cdot \varepsilon}{2d}.$$

Let $\mathcal{D}_{\Sigma,\varepsilon,t}$ be the distribution over $[2d]^t$ generated by selecting a vector $\sigma \in \Sigma$ uniformly at random, and then taking t i.i.d. samples from $\mathcal{D}_{\sigma,\varepsilon}$. Let $\mathcal{D}_{U,t} = \mathrm{U}([2d])^t$ be the distribution over $[2d]^t$

generated by selecting t i.i.d. uniform samples from $[2d]$. *Then* $\mathsf{TV}(\mathcal{D}_{U,t}, \mathcal{D}_{\Sigma,\varepsilon,t}) \leq f_{\mathsf{Paninski}}(t, \varepsilon, d)$ *for*

$$f_{\mathsf{Paninski}}(t, \varepsilon, d) = \frac{1}{2} \cdot \left(\exp\left(\frac{t^2\varepsilon^4}{d}\right) - 1\right)^{1/2}.$$

The proof also uses the following well-known fact about maximal couplings (see e.g. Lemma 4.1.13 in Roch, 2023+).

Theorem 13 *Let Ω be a set, and let $p_X, p_Y \in \Delta(\Omega)$ be distributions. Then*

$$\mathsf{TV}(p_X, p_Y) = \inf\Big\{\mathbb{P}[X \neq Y] : (X, Y) \textit{ is a joint distribution with marginals } X \sim p_X \textit{ and } Y \sim p_Y\Big\}.$$

Proof of Theorem 8 Let $X = \{x_1, \dots, x_d\} \subseteq \mathcal{X}$ be a set of size d that is shattered by $\mathcal{H}$ (such a set exists because $\mathsf{VC}(\mathcal{H}) = d$). Let $\mathcal{D}_U = \mathrm{U}(X \times \{0,1\})$.

For every $h \in \mathcal{H}_X = \{0,1\}^X$, let $\mathcal{D}_{h,4\varepsilon} \in \Delta(X \times \{0,1\})$ be a distribution such that

$$\forall\, (x,y) \in X \times \{0,1\} : \ \mathcal{D}_{h,4\varepsilon}\big((x,y)\big) = \begin{cases} (1+4\varepsilon)/2d & h(x) = y \\ (1-4\varepsilon)/2d & h(x) \neq y \end{cases}.$$

Consider a (possibly randomized) testing algorithm T that takes t i.i.d. samples from an unknown distribution $\mathcal{D}$ and decides correctly with probability at least $1-\beta$ whether $\mathcal{D} = \mathcal{D}_U$ or whether $\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : \ h \in \mathcal{H}_X\}$ (if $\mathcal{D}$ is not one of these $|\mathcal{H}_X| + 1$ options then we make no assumptions regarding the behavior of T).

Let $\mathcal{D}_{U,t} = (\mathcal{D}_U)^t$ and let $\mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}$ be the distribution generated by selecting $h \in \mathcal{H}_X$ uniformly at random and then taking t i.i.d. samples from $\mathcal{D}_{h,4\varepsilon}$. By Theorem 12, $\mathsf{TV}(\mathcal{D}_{U,t}, \mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}) \leq f_{\mathsf{Paninski}}(t, 4\varepsilon, d)$. By Theorem 13, for every $\alpha > 0$ there exists a joint distribution $(S_U, S_\mathcal{H})$ such that $S_U \sim \mathcal{D}_{U,t}$, $S_\mathcal{H} \sim \mathcal{D}_{\mathcal{H}_X,4\varepsilon,t}$, and $\mathbb{P}[S_U \neq S_\mathcal{H}] \leq f_{\mathsf{Paninski}}(t, 4\varepsilon, d) + \alpha$.

For any such α and $(S_U, S_\mathcal{H})$, no tester can distinguish with probability strictly greater than $1/2$ between S_U and $S_\mathcal{H}$ in the event where $S_U = S_\mathcal{H}$. Hence,

$$\beta \geq 1/2 \cdot \mathbb{P}[S_U = S_\mathcal{H}] = 1/2 \cdot (1 - \mathbb{P}[S_U \neq S_\mathcal{H}]) \geq 1/2 \cdot (1 - f_{\mathsf{Paninski}}(t, 4\varepsilon, d) - \alpha).$$

Taking $\alpha \to 0$ and rearranging yields

$$t \geq \frac{\sqrt{d \cdot \ln(1 + (4\beta - 2)^2)}}{\varepsilon^2}. \tag{2}$$

This establishes a lower bound on the sample complexity for the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : \ h \in \mathcal{H}_X\}$ distribution testing problem.

Next, we show a reduction from the distribution testing problem to PAC verification of $\mathcal{H}$. Let (V, P) be an interactive proof system that PAC verifies $\mathcal{H}$ such that the verifier V and honest prover P use m_V and m_P i.i.d. samples from the unknown distribution respectively, and satisfy Theorem 7 with parameters ε and δ, as in the statement of Theorem 8. Using (V, P), we construct a tester T for the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : \ h \in \mathcal{H}_X\}$ testing problem. Given sample access to an unknown distribution $\mathcal{D}$ for the testing problem, T operates as follows:

1. Compute $h_V = [V(\mathcal{D}), P(\mathcal{D}_U)]$. Namely, simulate an execution of the PAC verification protocol as follows. Take a sample $S_V \sim \mathcal{D}^{m_V}$ of m_V i.i.d. samples from $\mathcal{D}$, and take a sample $S_P \sim (\mathcal{D}_U)^{m_P}$ of m_P i.i.d. samples from $\mathcal{D}_U$ (seeing as the specification of $\mathcal{D}_U$ is completely known to T, T can generate as many samples from $\mathcal{D}_U$ as necessary using uniform random coins). Execute the PAC verification protocol such that V receives input S_V, P receives input S_P, and the output of the verifier at the end of the protocol is $h_V \in \mathcal{H} \cup \{\mathsf{reject}\}$.
2. Take a sample $S_{\mathsf{test}} \sim \mathcal{D}^\ell$ of $\ell = \lceil \ln(24)/2\varepsilon^2 \rceil < 3/\varepsilon^2$ i.i.d. samples from $\mathcal{D}$.
3. If $(h_V = \mathsf{reject}) \vee (h_V \neq \mathsf{reject} \wedge L^{0\text{-}1}_{S_{\mathsf{test}}}(h_V) \leq {}^1\!/\!_2 - 2\varepsilon)$ then output "$\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$". Otherwise, output "$\mathcal{D} = \mathcal{D}_U$".

We argue that the tester T defined in this manner solves the testing problem correctly with probability at least $7/12$. If $\mathcal{D} = \mathcal{D}_U$, then $L^{0\text{-}1}_{\mathcal{D}}(h) = 1/2$ for any $h \in \mathcal{H}$. In particular, if $h_V \neq \mathsf{reject}$ then $L^{0\text{-}1}_{S_{\mathsf{test}}}(h_V) \geq 1/2 - \varepsilon$ with probability at least $11/12$ (by Hoeffding's inequality and the choice of ℓ). Thus, if $\mathcal{D} = \mathcal{D}_U$ then T outputs "$\mathcal{D} = \mathcal{D}_U$" with probability at least $11/12$.

Conversely, if $\mathcal{D} = \mathcal{D}_{h',4\varepsilon}$ for some $h' \in \mathcal{H}_X$, then $L^{0\text{-}1}_{\mathcal{D}}(h) = 1/2 - 4\varepsilon$ for $h \in \mathcal{H}$ such that $h|_X = h'$. From the correctness of the PAC verification protocol, with probability at least $2/3$, either $h_V = \mathsf{reject}$, or $L^{0\text{-}1}_{\mathcal{D}}(h_V) \leq 1/2 - 3\varepsilon$, and in that case with probability at least $11/12$, $L^{0\text{-}1}_{S_{\mathsf{test}}}(h) \leq 1/2 - 2\varepsilon$ (again by Hoeffding's inequality and choice of ℓ). A union bound implies that if $\mathcal{D} = \mathcal{D}_{h',4\varepsilon}$ for some $h' \in \mathcal{H}_X$ then T outputs "$\mathcal{D} \in \{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$" with probability at least $1 - 1/3 - 1/12 = 7/12$.

We conclude that T correctly solves the $\mathcal{D}_U$ vs. $\{\mathcal{D}_{h,4\varepsilon} : h \in \mathcal{H}_X\}$ testing problem with probability at least $7/12$ using $t = m_V + \ell$ i.i.d. samples from the unknown distribution $\mathcal{D}$. Plugging $\beta = 5/12$ in Eq. (2), this implies that $m_V \geq (0.3 \cdot \sqrt{d} - 3)/\varepsilon^2$, as desired. ■

Remark 14 *A previous version of this paper (Mutreja and Shafer, 2022) presented a proof of an $\Omega\left(\sqrt{d}\right)$ lower bound, without the dependence on ε. That proof uses a reduction to a simpler distribution testing lower bound based on the 'birthday paradox' (instead of the Paninski bound), and it may be better suited for pedagogical expositions.*

4. Verification of Unions of Intervals

Theorem 15 (Canonne et al. 2022, Theorem 1[4]) *Let $\varepsilon, \delta \in (0,1)$, let $n \in \mathbb{N}$, and let $\mathcal{P}, \tilde{\mathcal{P}} \in \Delta([n])$ be distributions. There exists a tolerant distribution identity tester that, given a complete description of $\tilde{\mathcal{P}}$ and $m = O\left(\sqrt{n}\log(1/\delta)\varepsilon^{-2}\right)$ i.i.d. samples from $\mathcal{P}$, satisfies the following:*

- ***Completeness.*** *If $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \leq \varepsilon/\sqrt{n}$ then the tester accepts with probability at least $1 - \delta$.*
- ***Soundness.*** *If $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) > \varepsilon$ then the tester rejects with probability at least $1 - \delta$.*

Definition 16 *Let $\varepsilon \in [0,1]$, let $\mathcal{X}$ be a set and let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ be a set of functions. Let $\mathcal{D} \in \Delta(\mathcal{X})$, and let $S \in \mathcal{X}^m$ for some $m \in \mathbb{N}$. We say that S is an ε-sample for $\mathcal{D}$ with respect to $\mathcal{F}$*

4. See also Goldreich and Ron (2011) and the discussion following Theorem 5.4 in Canonne (2020a).

if

$$\forall f \in \mathcal{F}: \left| \frac{|\{x \in S: f(x) = 1\}|}{m} - \mathbb{P}_{x \sim \mathcal{D}}[f(x) = 1] \right| \leq \varepsilon.$$

Theorem 17 **(Vapnik and Chervonenkis, 1968[5])** *Let $d \in \mathbb{N}$ and $\varepsilon, \delta \in (0,1)$. Let $\mathcal{X}$ be a set and let $\mathcal{F} \subseteq \{0,1\}^{\mathcal{X}}$ be a set of functions with $\mathsf{VC}(\mathcal{F}) = d$. Let $\mathcal{D} \in \Delta(\mathcal{X})$, and let $S \sim \mathcal{D}^m$, where*

$$m = \Omega\left(\frac{d \log(d/\varepsilon) + \log(1/\delta)}{\varepsilon^2}\right).$$

Then with probability at least $1 - \delta$, S is an ε-sample for $\mathcal{D}$ with respect to $\mathcal{F}$.

Proof of Theorem 9 We show that Protocol 1 (in Appendix A) satisfies the requirements of the theorem. For completeness, note that if the prover follows the protocol then $\tilde{P}_{j,0} + \tilde{P}_{j,1} = 1/k$ for all j, so the verifier will never reject at the first 'if' statement. Let $\mathcal{B} = \{I_j \times \{y\}: j \in [k] \wedge y \in \{0,1\}\}$, and let $\mathcal{F} = \{\mathbb{1}_E : E \in \sigma(\mathcal{B})\} \subseteq \{0,1\}^{[0,1] \times \{0,1\}}$. In words, $\mathcal{F}$ is the set of indicator functions for events in the σ-algebra generated by $\mathcal{B}$. $\mathsf{VC}(\mathcal{F}) = 2k = O(d/\varepsilon)$, so Theorem 17 and the choice of m_P imply that with probability at least $1 - \delta/2$, S_P is an $\varepsilon/(6\sqrt{2k})$-sample for $\mathcal{D}$ with respect to $\mathcal{F}$. By the definitions of total variation distance and of an ε-sample, this implies that $\mathbb{P}\left[\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \leq \varepsilon/(6\sqrt{2k})\right] \geq 1 - \delta/2$. From the completeness of the tester of Theorem 15 and a union bound we conclude that with probability at least $1 - \delta$, the verifier does not reject. This establishes completeness.

For soundness, consider two cases.

- The prover is too dishonest, such that $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) > \varepsilon/6$. Then by the soundness of the tester of Theorem 15, the verifier rejects with probability at least $1 - \delta/2$.
- The prover is sufficiently honest, such that $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \leq \varepsilon/6$. Then for any $h' \in \mathcal{H}_d$,

$$\begin{aligned} \left| L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\tilde{\mathcal{P}}}^{0\text{-}1}(h') \right| &\leq \left| L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\mathcal{P}}^{0\text{-}1}(h') \right| + \left| L_{\mathcal{P}}^{0\text{-}1}(h') - L_{\tilde{\mathcal{P}}}^{0\text{-}1}(h') \right| \\ &\leq \left| L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\mathcal{P}}^{0\text{-}1}(h') \right| + \varepsilon/6, \end{aligned} \tag{3}$$

  where the last inequality follows from $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \leq \varepsilon/6$.

  Fix $h' \in \mathcal{H}_d$. We argue that $\left| L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\mathcal{P}}^{0\text{-}1}(h') \right| \leq \varepsilon/3$. Let $Q = \{x \in [0,1]: h'(x) \neq h'(x^*)\}$, where for each $x \in [0,1]$, we define $x^* = x_j^*$ such that $x \in I_j$. Namely, Q is the set of points for which applying the discretization procedure alters the output of h'. Then

$$\begin{aligned} \left| L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\mathcal{P}}^{0\text{-}1}(h') \right| &= \left| \mathbb{P}_{(x,y) \sim \mathcal{D}}\left[h'(x) \neq y\right] - \mathbb{P}_{(x,y) \sim \mathcal{D}}\left[h'(x^*) \neq y\right] \right| \\ &= \Big| \mathbb{P}_{(x,y) \sim \mathcal{D}}\left[h'(x) \neq y \wedge x \in Q\right] \\ &\qquad - \mathbb{P}_{(x,y) \sim \mathcal{D}}\left[h'(x^*) \neq y \wedge x \in Q\right] \Big| \end{aligned} \tag{4}$$

5. Cf. Alon and Spencer (2000), Theorem 13.4.4.

$$
\begin{aligned}
&\leq \mathcal{D}(Q') && (Q' = Q \times \{0,1\}) \\
&\leq \sum_{j\in[k]:\ I_j\cap Q\neq\varnothing} \mathcal{D}(I_j') && (I_j' = I_j \times \{0,1\}) \\
&= \sum_{j\in[k]:\ I_j\cap Q\neq\varnothing} \mathcal{P}(I_j') && (\mathcal{D}(I_j') = \mathcal{P}(I_j')) \\
&= \mathcal{P}\left(\bigcup\left\{I_j' :\ I_j \cap Q \neq \varnothing\right\}\right) \\
&\leq \tilde{\mathcal{P}}\left(\bigcup\left\{I_j' :\ I_j \cap Q \neq \varnothing\right\}\right) + \mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \\
&\leq 2d/k + \mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) && (5) \\
&\leq 2d/k + \varepsilon/6 = \varepsilon/3, && (6)
\end{aligned}
$$

where Eq. (4) holds since the loss of h' can differ between $\mathcal{D}$ and $\mathcal{P}$ only for points in Q; Eq. (5) holds because h' consists of d intervals, which together have $2d$ endpoints, $I_j \cap Q \neq \varnothing$ only if I_j contains one of these endpoints, and if the verifier did not reject then $\tilde{\mathcal{P}}(I_j') = {}^1\!/k$ for all j; finally Eq. (6) holds by the assumption (in the current case) that the prover is sufficiently honest.

Combining Eq. (6) with Eq. (3) yields $\forall h' \in \mathcal{H}_d :\ \left|L_{\mathcal{D}}^{0\text{-}1}(h') - L_{\tilde{\mathcal{P}}}^{0\text{-}1}(h')\right| \leq \varepsilon/2$. This implies that a hypothesis h that has minimum loss with respect to $\tilde{\mathcal{P}}$ satisfies $L_{\mathcal{D}}^{0\text{-}1}(h) \leq L_{\mathcal{D}}^{0\text{-}1}(\mathcal{H}) + \varepsilon$.

We conclude that regardless of the prover's behavior, with probability at least $1 - \delta/2$ the verifier either rejects or outputs a hypothesis with excess loss at most ε, as desired. ■

Remark 18 *The dependence of the tolerance parameter in Theorem 15 on the domain size is quadratic, namely the verifier accepts if $\mathsf{TV}\left(\mathcal{P}, \tilde{\mathcal{P}}\right) \leq \varepsilon/\sqrt{n}$. Notice that this affects the sample complexity of the honest prover but not of the verifier. For instance, if the tolerance was ε/e^n instead of $\varepsilon/\sqrt{n}$, the verifier's sample complexity would remain unchanged.*

5. Discussion and Future Work

In this paper, we have shown that $\Omega\left(\sqrt{d}\right)$ samples are necessary for PAC verifying a class of VC dimension d, and furthermore, for some classes $O\left(\sqrt{d}\right)$ samples are sufficient. In contrast, Lemma 4.1 in Goldwasser et al. (2021) states that there also exist VC classes where the sample complexity for verification is $\tilde{\Omega}(d)$ under the assumption that the verifier is proper (outputs a hypothesis from the class), and we believe it is likely that there exist VC classes for which an $\tilde{\Omega}(d)$ lower bound holds for any verifier.

Hence, it appears likely that the VC dimension does not characterize the sample complexity of PAC verification. In that case, finding an alternative combinatorial quantity that does characterize that sample complexity is an exciting open problem.

A potentially easier problem is to devise upper bounds (PAC verification protocols) for specific classes of interest. For example, the main property of the thresholds class utilized in the proof of

Theorem 9 is that it has low 'surface area' or noise sensitivity (cf. Balcan et al., 2012). Perhaps a similar proof technique could apply to other classes as well.

Additionally, we introduced a notion of PAC verification of an algorithm. We believe this is very natural definition, because many of the algorithms that people might like to delegate in practice are not PAC learners, including unsupervised learning algorithms (e.g., clustering and dimensionality reduction algorithms), and supervised algorithms that are not provably PAC learners (e.g., neural networks trained via SGD). Devising PAC verification protocols for specific algorithms of interest could be a rewarding endeavor.

Acknowledgments

An initial version of the lower bound in Theorem 8 resulted from a conversation with Lijie Chen and Guy Rothblum. JS would like to thank Shafi Goldwasser, Steve Hanneke, Bobby Kleinberg, Shay Moran, Ido Nachum, Guy Rothblum and Abhishek Shetty for helpful comments and suggestions. Part of this work was done while JS was visiting the Weizmann Institute of Science (hosted by Guy Rothblum), Cornell University (hosted by Bobby Kleinberg) and the Technion (hosted by Shay Moran). JS is grateful for their hospitality and support.

This work was supported in part by DARPA (Defense Advanced Research Projects Agency) contract #HR001120C0015, and the Simons Collaboration on the Theory of Algorithmic Fairness. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Simons Foundation or DARPA.

References

Noga Alon and Joel H. Spencer. *The Probabilistic Method, Second Edition*. John Wiley, 2000. ISBN 978-0-47137046-8. doi: 10.1002/0471722154. URL https://doi.org/10.1002/0471722154.

Cem Anil, Guodong Zhang, Yuhuai Wu, and Roger B. Grosse. Learning to give checkable answers with prover-verifier games. *CoRR*, abs/2108.12099, 2021. URL https://arxiv.org/abs/2108.12099.

Maria-Florina Balcan, Eric Blais, Avrim Blum, and Liu Yang. Active property testing. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012*, pages 21–30. IEEE Computer Society, 2012. doi: 10.1109/FOCS.2012.64. URL https://doi.org/10.1109/FOCS.2012.64.

Ran Canetti and Ari Karchmer. Covert learning: How to learn with an untrusted intermediary. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2021. doi: 10.1007/978-3-030-90456-2_1. URL https://doi.org/10.1007/978-3-030-90456-2_1.

Clément L. Canonne. A survey on distribution testing: Your data is big. But is it blue? *Theory of Computing*, pages 1–100, 2020a. doi: dx.doi.org/10.4086/toc.gs.2020.009. URL http://dx.doi.org/10.4086/toc.gs.2020.009.

Clément L. Canonne, Ayush Jain, Gautam Kamath, and Jerry Li. The price of tolerance in distribution testing. In Po-Ling Loh and Maxim Raginsky, editors, *Conference on Learning Theory, 2-5 July 2022, London, UK*, volume 178 of *Proceedings of Machine Learning Research*, pages 573–624. PMLR, 2022. URL https://proceedings.mlr.press/v178/canonne22a.html.

Clément L. Canonne. A short note on learning discrete distributions. abs/2002.11457, 2020b. doi: 10.48550/ARXIV.2002.11457. URL https://doi.org/10.48550/arXiv.2002.11457.

Matthias C. Caro, Marcel Hinsche, Marios Ioannou, Alexander Nietner, and Ryan Sweke. Classical verification of quantum learning. *CoRR*, abs/2306.04843, 2023. doi: 10.48550/arXiv.2306.04843. URL https://doi.org/10.48550/arXiv.2306.04843.

Alessandro Chiesa and Tom Gur. Proofs of proximity for distribution testing. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 53:1–53:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi: 10.4230/LIPIcs.ITCS.2018.53. URL https://doi.org/10.4230/LIPIcs.ITCS.2018.53.

Oded Goldreich and Dana Ron. On testing expansion in bounded-degree graphs. In Oded Goldreich, editor, *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation - In Collaboration with Lidor Avigad, Mihir Bellare, Zvika Brakerski, Shafi Goldwasser, Shai Halevi, Tali Kaufman, Leonid Levin, Noam Nisan, Dana Ron, Madhu Sudan, Luca Trevisan, Salil Vadhan, Avi Wigderson, David Zuckerman*, volume 6650 of *Lecture Notes in Computer Science*, pages 68–75. Springer, 2011. doi: 10.1007/978-3-642-22670-0_9. URL https://doi.org/10.1007/978-3-642-22670-0_9.

Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive proofs for verifying machine learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 41:1–41:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi: 10.4230/LIPIcs.ITCS.2021.41. URL https://doi.org/10.4230/LIPIcs.ITCS.2021.41.

David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992. doi: 10.1016/0890-5401(92)90010-D. URL https://doi.org/10.1016/0890-5401(92)90010-D.

Tal Herman and Guy N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1208–1219. ACM, 2022. doi: 10.1145/3519935.3519987. URL https://doi.org/10.1145/3519935.3519987.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, pages 13–30, 1963. doi: doi.org/10.2307/2282952. URL https://doi.org/10.2307/2282952.

Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998. doi: 10.1145/293347.293351. URL https://doi.org/10.1145/293347.293351.

Saachi Mutreja and Jonathan Shafer. PAC verification of statistical algorithms. *CoRR*, abs/2211.17096, 2022. URL https://arxiv.org/abs/2211.17096v1.

Liam Paninski. A coincidence-based test for uniformity given very sparsely sampled discrete data. *IEEE Trans. Inf. Theory*, 54(10):4750–4755, 2008. doi: 10.1109/TIT.2008.928987. URL https://doi.org/10.1109/TIT.2008.928987.

Sebastien Roch. *Modern Discrete Probability: an Essential Toolkit*. Forthcoming, Cambridge University Press, 2023+. URL https://people.math.wisc.edu/~roch/mdp/.

Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. *CoRR*, abs/2204.07196, 2022. doi: 10.48550/arXiv.2204.07196. URL https://doi.org/10.48550/arXiv.2204.07196.

Sanjit A. Seshia, Dorsa Sadigh, and S. Shankar Sastry. Toward verified artificial intelligence. *Commun. ACM*, 65(7):46–55, 2022. doi: 10.1145/3503914. URL https://doi.org/10.1145/3503914.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014. ISBN 978-1-10-705713-5. doi: doi.org/10.1017/CBO9781107298019. URL https://doi.org/10.1017/CBO9781107298019.

Leslie G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984. doi: 10.1145/1968.1972. URL https://doi.org/10.1145/1968.1972.

V. N. Vapnik and A. Ya. Chervonenkis. The uniform convergence of frequencies of the appearance of events to their probabilities. *Dokl. Akad. Nauk SSSR*, 181(4):781–783, 1968. URL http://www.ams.org/mathscinet-getitem?mr=0231431.

Bin Yu. Assouad, Fano, and Le Cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer, 1997. doi: doi.org/10.1007/978-1-4612-1880-7_29. URL https://doi.org/10.1007/978-1-4612-1880-7_29.

Appendix A. Protocol for Unions of Intervals

Assumptions:
- $d, {}^1\!/\!_\varepsilon \in \mathbb{N}$ (this can always be achieved by making ε smaller if necessary), $k = 12d/\varepsilon$.
- $m_P = O\Big(\big(d^2 \log(d/\varepsilon) + \log(1/\delta)\big)\varepsilon^{-4}\Big)$ is a multiple of k.
- $m_V = O\Big(\sqrt{d}\log(1/\delta)\varepsilon^{-2.5}\Big)$.
- $S_V \sim \mathcal{D}^{m_V}$, $S_P \sim \mathcal{D}^{m_P}$.
- $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ is an unknown target distribution.

PROVER$(S_P, \delta, \varepsilon)$:

$I_1, I_2, \ldots, I_k \leftarrow$ a partition of $[0,1]$ into disjoint intervals such that $\cup_{i \in [k]} I_i = [0,1]$ and $\forall j \in [k] : \ |\{x_1^P, \ldots, x_{m_P}^P\} \cap I_j| = m_P/k$.

for $j \in [k]$:
  for $b \in \{0,1\}$:
    $\tilde{P}_{j,b} \leftarrow |\{(x,y) \in S_P : \ x \in I_j \ \wedge \ y = b\}|/m_P$ ▷ Counted as a multiset

send $(I_1, \ldots, I_k)$ and $\big(\tilde{P}_{j,y}\big)_{j \in [k], y \in \{0,1\}}$ to the verifier

VERIFIER$(S_V, \delta, \varepsilon)$:

receive $(I_1, \ldots, I_k)$ and $\big(\tilde{P}_{j,y}\big)_{j \in [k], y \in \{0,1\}}$ from the prover

if $\exists j \in [k]$ s.t. $\tilde{P}_{j,0} + \tilde{P}_{j,1} \neq {}^1\!/\!_k$:
  output reject and **terminate**

$x_1^*, \ldots, x_k^* \leftarrow$ arbitrary points such that $\forall j \in [k] : \ x_j^* \in I_j$

execute the tester of Theorem 15 with parameters ${}^\varepsilon\!/\!_6$, ${}^\delta\!/\!_2$ where $\mathcal{P}, \tilde{\mathcal{P}} \in \Delta([0,1] \times \{0,1\})$ are as follows:
- $\mathcal{P}$ is the distribution generated by sampling $(x,y) \sim \mathcal{D}$ and then outputting (x^*, y) where $x^* = x_j^*$ such that $x \in I_j$
- $\tilde{\mathcal{P}}$ is the distribution such that $\mathbb{P}\Big[(x_j^*, y)\Big] = \tilde{P}_{j,y}$ for all $j \in [k], y \in \{0,1\}$

if distribution identity tester rejects:
  output reject and **terminate**

$h \leftarrow \operatorname{argmin}_{h' \in \mathcal{H}_d} L_{\tilde{\mathcal{P}}}^{0\text{-}1}(h')$

output h

Protocol 1: Verification protocol for unions of d-intervals.

Appendix B. Verification of Statistical Query Algorithms

B.1. Definitions

B.1.1. Statistical Query Algorithms

Definition 19 **(Kearns, 1998)** *Let Ω be a set, let $\mathcal{D} \in \Delta(\Omega)$ be a distribution, and let $\tau \geq 0$. A statistical query is an indicator function $q : \Omega \to \{0, 1\}$. An oracle $\mathcal{O}$ is a statistical query oracle for $\mathcal{D}$ with precision τ, denoted $\mathcal{O} \in \mathsf{SQ}(\mathcal{D}, \tau)$, if at each invocation, $\mathcal{O}$ takes a statistical query q as input and produces an arbitrary evaluation $\mathcal{O}(q) \in [0, 1]$ as output such that*

$$\left|\mathcal{O}(q) - \mathbb{E}_{X\sim\mathcal{D}}[q(X)]\right| \leq \tau. \tag{7}$$

In particular, the oracle's evaluations may be adversarial and adaptive, as long as each of them satisfies Eq. (7).

Remark 20 *The notion of PAC verification of an algorithm (Theorem 10) requires that the verifier's output be competitive with $L_{\mathcal{D}}(A) = \mathbb{E}\left[L_{\mathcal{D}}\left(A^{\mathcal{O}}\right)\right]$, the expected loss of algorithm A when executed with access to oracle $\mathcal{O}$. For this expectation to be defined, throughout this paper we only consider oracles whose behavior can be described by a probability measure. In particular, oracles may be adaptive and adversarial in a deterministic or randomized manner, but they cannot be arbitrary.*

Definition 21 *A statistical query algorithm is a (possibly randomized) algorithm A that takes no inputs and has access to a statistical query oracle $\mathcal{O}$. At each time step $t = 1, 2, 3, \ldots$:*

- *A chooses a finite batch $\mathbf{q}_t = \left(q_t^1, \ldots, q_t^{n_t}\right)$ of statistical queries and sends it to the oracle $\mathcal{O}$.*
- *$\mathcal{O}$ sends a batch of evaluations $\mathbf{v}_t = \left(v_t^1, \ldots, v_t^{n_t}\right) \in [0, 1]^{n_t}$ to A, such that $v_t^i = \mathcal{O}(q_t^i)$ for all $i \in [n_t]$.*
- *A either produces an output and terminates, or continues to time step $t + 1$.*

The resulting sequence $\mathbf{r} = (\mathbf{q}_1, \mathbf{v}_1, \mathbf{q}_2, \mathbf{v}_2, \ldots)$ is called a transcript of the execution.

Note that for each t, the choice of $\mathbf{q}_t$ is a deterministic function of $(\mathbf{r}_{<t}, \rho)$, where

$$\mathbf{r}_{<t} = (\mathbf{q}_1, \mathbf{v}_1, \mathbf{q}_2, \mathbf{v}_2, \ldots, \mathbf{q}_{t-1}, \mathbf{v}_{t-1}),$$

and ρ denotes the randomness of A. If A terminates, its final output is a deterministic function of $(\mathbf{r}, \rho)$.

B.1.2. The Partition Size

Definition 22 *Let Ω be a set, and let $\mathcal{S} \subseteq 2^{\Omega}$ be a collection of subsets. We say that $\mathcal{S}$ is a σ-algebra for Ω if it satisfies the following properties:*

- *$\Omega \in \mathcal{S}$.*
- *$\forall S \subseteq \mathcal{S} : \ \Omega \setminus S \in \mathcal{S}$.*
- *For any countable sequence $S_1, S_2, \ldots \in \mathcal{S} : \ \cup_{i=1}^{\infty} S_i \in \mathcal{S}$.*

Definition 23 *Let Ω be a set.*

- *Let $\mathcal{A} \subseteq 2^{\Omega}$ be a collection of subsets. The σ-algebra generated by $\mathcal{A}$ for Ω, denoted $\sigma(\mathcal{A})$, is the intersection of all σ-algebras for Ω that are supersets of $\mathcal{A}$.*
- *Let $\mathcal{F} \subseteq \{0,1\}^{\Omega}$ be a set of indicator functions. The σ-algebra generated by $\mathcal{F}$ for Ω is $\sigma(\mathcal{F}) = \sigma\left(\{A \subseteq \Omega : \mathbb{1}_A \in \mathcal{F}\}\right)$.*

Definition 24 *Let $\mathcal{S}$ be a σ-algebra. The set of atoms of $\mathcal{S}$ is*

$$\mathsf{Atoms}(\mathcal{S}) = \left\{S \in \mathcal{S} : \ \left(\forall S' \in \mathcal{S} \setminus \varnothing : \ S' \not\subset S\right)\right\}.$$[6]

Definition 25 *Let Ω be a set and let $\mathcal{F} = \{f_1, f_2, \dots, f_k\} \subseteq \{0,1\}^{\Omega}$ be a finite set of indicator functions. The partition size of $\mathcal{F}$ is $\mathsf{PS}(\mathcal{F}) = |\mathsf{Atoms}(\sigma(\mathcal{F}))| \in \mathbb{N}$, i.e., the number of atoms in the σ-algebra generated by $\mathcal{F}$ for Ω.*

B.2. Formal Statements

Theorem 26 (PAC Verification of an SQ Algorithm) *Let $b, s \in \mathbb{N}$, let Ω be a set and $\mathcal{H}$ be a discrete set. Let A be a statistical query algorithm that adaptively and randomly generates some random number T of batches $\mathbf{q}_1, \dots, \mathbf{q}_T$ of statistical queries $\Omega \to \{0,1\}$ such that with probability 1, $T \leq b$ and $\mathsf{PS}(\mathbf{q}_t) \leq s$ for each $t \in [T]$, and the algorithm outputs a random value $h \in \mathcal{H}$. Let $\mathbb{D} \subseteq \Delta(\Omega)$ be a set of distributions, let $\tau > 0$, and let $L : \Omega \times \mathcal{H} \to [0,1]$ be a loss function.*

Then there exists a collection of oracles $\mathbb{O} = \{\mathcal{O}_{\mathcal{D}}\}_{\mathcal{D} \in \mathbb{D}}$ where $\mathcal{O}_{\mathcal{D}} \in \mathsf{SQ}(\mathcal{D}, \tau)$ for all $\mathcal{D} \in \mathbb{D}$, such that algorithm A with access to oracles $\mathbb{O}$ is PAC verifiable with respect to $\mathbb{D}$ by a verification protocol that uses random samples, where the verifier and honest prover respectively use

$$m_V = \Theta\left(\frac{\sqrt{s}\log(bk/\delta)}{\tau^2} + \frac{\log(k/\delta)}{\varepsilon^2}\right),$$

and

$$m_P = \Theta\left(\frac{s^3\log(sbk/\delta\tau)}{\tau^2}\right)$$

i.i.d. samples, with $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.

As a corollary, we obtain that for statistical query algorithms of a particular type, the sample complexity of PAC verification has a quadratically lower dependence on the VC dimension of the batches of statistical queries compared to simulating the algorithm using random samples.

Corollary 27 *Let A be a statistical query algorithm as in Theorem 26, and let $d \in \mathbb{N}$. Assume that in each time step $t \in [T]$, $\mathsf{VC}(\mathbf{q}_t) = d$ and $|\mathbf{q}_t| = 2^d$. Namely, $\mathbf{q}_t$ is the set of indicator functions of a σ-algebra with d atoms. Consider an implementation of A that uses random samples to simulate the SQ oracle accessed by A, such that the implementation uses random samples only and does not use any oracles. Simulating an oracle $\mathcal{O} \in \mathsf{SQ}(\mathcal{D}, \tau)$ requires*

$$m = \Omega\left(\frac{d + \log(1/\delta)}{\tau^2}\right)$$

6. $S' \not\subset S$ denotes that S' is not a strict subset of S.

i.i.d. samples from $\mathcal{D}$. In contrast, there exists a protocol that PAC verifies A such that the verifier uses only

$$m_V = \Theta\left(\frac{\sqrt{d}\log(bk/\delta)}{\tau^2} + \frac{\log(k/\delta)}{\varepsilon^2}\right)$$

i.i.d. samples from $\mathcal{D}$, with $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.

The lower bound in the corollary is the standard VC lower bound.

B.3. Proofs

Definition 28 *Let A be a statistical query algorithm, let $\mathbb{D}$ be a collection of distributions, and let $\varepsilon, \tau > 0$. We say that a collection of oracles $\mathbb{O} = \{\mathcal{O}_\mathcal{D}\}_{\mathcal{D}\in\mathbb{D}}$ is $\underline{\varepsilon\text{-maximizing with respect to } A \text{ and } \mathbb{D}}$ if for each $\mathcal{D} \in \mathbb{D}$, $\mathcal{O}_\mathcal{D} \in \mathsf{SQ}(\mathcal{D}, \tau)$ and $\mathbb{E}\left[L_\mathcal{D}\left(A^{\mathcal{O}_\mathcal{D}}\right)\right] \geq \sup_{\mathcal{O}\in\mathsf{SQ}(\mathcal{D},\tau)} \mathbb{E}\left[L_\mathcal{D}\left(A^{\mathcal{O}}\right)\right] - \varepsilon$.*

Proof of Theorem 26 Fix a collection of oracles $\mathbb{O} = \{\mathcal{O}_\mathcal{D}\}_{\mathcal{D}\in\mathbb{D}}$ that is $\varepsilon/4$-maximizing with respect to A and $\mathbb{D}$. We show that algorithm A with access to the oracles $\mathbb{O}$ is PAC verified by Protocol 2.

To establish completeness, notice that each batch $\mathbf{a}_t$ of queries sent to the prover by VERIFIERITERATION satisfies $\mathsf{VC}(\mathbf{a}_t) = 1$, and there are at most $b \cdot k$ such batches. Hence, by Theorem 17 and a union bound, taking m_P as in the statement is sufficient to guarantee that with probability at least $1 - \delta/4$,

$$\forall \text{ iteration } i \in [k]\ \forall t \in [T] :\ \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_\infty \leq \frac{\tau}{s\sqrt{s}},$$

where $\mathbf{p}_t$ is the vector of correct evaluations, with components $p_t^j = \mathbb{E}_{Z\sim\mathcal{D}}\left[a_t^j(Z)\right]$. Hence, with probability at least $1 - \delta/4$,

$$\forall \text{ iteration } i \in [k]\ \forall t \in [T] :\ \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \frac{\tau}{\sqrt{s}}. \tag{8}$$

By Eq. (8), Theorem 15, and the choice of m_V, with probability at least $1 - \delta/4$, none of the executions of IDENTITYTEST cause the verifier to reject.

By a union bound, with probability at least $1 - \delta/2$, Eq. (8) holds and the verifier does not reject. Then, by Theorem 29,

$$\forall i \in [k] :\ \mathbb{P}\left[L_\mathcal{D}(h_i) \leq L_\mathcal{D}(A) + \frac{\varepsilon}{2}\right] \geq \frac{\varepsilon}{8}. \tag{9}$$

By the choice of k,

$$\mathbb{P}\left[\forall i \in [k] :\ L_\mathcal{D}(h_i) > L_\mathcal{D}(A) + \frac{\varepsilon}{2}\right] \leq \left(1 - \frac{\varepsilon}{8}\right)^k \leq e^{-\varepsilon k/8} \leq \frac{\delta}{4}. \tag{10}$$

By Hoeffding's inequality, a union bound, and the choice of m_V,

$$\mathbb{P}\left[\forall i \in [k] :\ \left|L_{S'_V}(h_i) - L_\mathcal{D}(h_i)\right| \leq \frac{\varepsilon}{2}\right] \geq 1 - \frac{\delta}{4}. \tag{11}$$

Combining Eqs. (8), (10) and (11) via a union bound, we conclude that with probability $1 - \delta$, the verifier does not reject and it outputs $h \in \mathcal{H}$ such that $L_\mathcal{D}(h) \leq L_\mathcal{D}(A) + \varepsilon$. This establishes completeness.

To establish soundness, consider an interaction between the verifier of Protocol 2 and any deterministic or randomized (possibly malicious and computationally unbounded) prover P', and examine the following two events.

- Event I: the evaluations provided by P' satisfy

$$\forall \text{ iteration } i \in [k] \; \forall t \in [T] : \; \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \tau. \tag{12}$$

  If the verifier does not reject then Theorem 29 implies that Eq. (9) holds. As we saw in the proof for the completeness requirement, this implies that with probability at least $1 - \delta$, the verifier outputs $h \in \mathcal{H}$ such that $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(A) + \varepsilon$.

- Event II: there exists an iteration $i \in [k]$ containing a time step $t^* \in [T]$ such that $\|\tilde{\mathbf{p}}_{t^*} - \mathbf{p}_{t^*}\|_1 > \tau$. By Theorem 15 and the choice of m_V, with probability at least $1 - \delta/4$ the verifier rejects in time step t^*.

We conclude that in both cases,

$$\mathbb{P}_{S_V \sim \mathcal{D}^{m_V}}[h = \mathsf{reject} \; \vee \; L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(A) + \varepsilon] \geq 1 - \delta,$$

and this establishes soundness. ■

Lemma 29 *In the context of Theorem 26, fix a distribution $\mathcal{D} \in \mathbb{D}$ and let $\mathcal{O}_{\mathcal{D}} \in \mathsf{SQ}(\mathcal{D}, \tau)$ be an oracle such that*

$$\mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_{\mathcal{D}}}\big)\big] \geq \sup_{\mathcal{O} \in \mathsf{SQ}(\mathcal{D},\tau)} \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}}\big)\big] - \varepsilon/4.$$

Consider an execution of VERIFIERITERATION *(Protocol 3). Let G denote the event in which the verifier does not reject, and the query evaluations $\tilde{\mathbf{p}}_t$ provided by the prover satisfy*

$$\forall t \in [T] : \; \|\tilde{\mathbf{p}}_t - \mathbf{p}_t\|_1 \leq \tau, \tag{13}$$

where $\mathbf{p}_t$ is the vector of correct evaluations $p_t^i = \mathbb{E}_{Z \sim \mathcal{D}}\big[a_t^i(Z)\big]$. Then the output $h_i \in \mathcal{H}$ returned by VERIFIERITERATION *satisfies*

$$\mathbb{P}\Big[L_{\mathcal{D}}(h_i) \leq \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_{\mathcal{D}}}\big)\big] + \frac{\varepsilon}{2} \;\Big|\; G\Big] \geq \frac{\varepsilon}{8}. \tag{14}$$

Proof Let $\mathcal{O}_G$ denote the oracle with evaluations that are equal in distribution to the evaluations provided by the prover conditioned on event G occurring. By the choice of $\mathcal{O}_{\mathcal{D}}$,

$$\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] = \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_G}\big)\big] \leq \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_{\mathcal{D}}}\big)\big] + \varepsilon/4.$$

By Markov's inequality,

$$\begin{aligned}
\mathbb{P}\big[L_{\mathcal{D}}(h_i) > \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_{\mathcal{D}}}\big)\big] + \varepsilon/2 \mid G\big] &\leq \mathbb{P}\big[L_{\mathcal{D}}(h_i) > \mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] + \varepsilon/4 \mid G\big] \\
&\leq \frac{\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G]}{\mathbb{E}[L_{\mathcal{D}}(h_i) \mid G] + \varepsilon/4} \\
&\leq \frac{1}{1 + \varepsilon/4},
\end{aligned}$$

since $L_{\mathcal{D}}$ is at most 1. Hence, the complement satisfies

$$\mathbb{P}\Big[L_{\mathcal{D}}(h_i) \leq \mathbb{E}\big[L_{\mathcal{D}}\big(A^{\mathcal{O}_{\mathcal{D}}}\big)\big] + \frac{\varepsilon}{2} \,\Big|\, G\Big] \leq \frac{\varepsilon/4}{1+\varepsilon/4} \leq \frac{\varepsilon}{8},$$

as desired. ■

Assumptions:

- Ω is a set, $\mathcal{D} \in \Delta(\Omega)$ is the population distribution.
- A is a statistical query algorithm to be verified.
- $\tau \in (0,1)$ is the accuracy parameter for statistical queries used by A.
- $b \in \mathbb{N}$ is an upper bound on the number of statistical query batches generated by A.
- $\varepsilon, \delta \in (0,1)$ are the desired accuracy and confidence parameters for the verification.
- $k = \lceil 8\log(4/\delta)/\varepsilon \rceil$.
- $m_V = \Theta\big(\sqrt{s}\log(bk/\delta)\tau^{-2} + \log(k/\delta)\varepsilon^{-2}\big)$.
- $m_P = \Theta\big(s^3\log(sbk/\delta\tau)\tau^{-2}\big)$.
- $S_V, S'_V \sim \mathcal{D}^{m_V}$, $S_P \sim \mathcal{D}^{m_P}$ are independent sets of i.i.d. samples.
- $S_V = (z_1^V, \dots, z_{m_V}^V)$, $S'_V = (z_1^{V'}, \dots, z_{m_V}^{V'})$, $S_P = (z_1^P, \dots, z_{m_P}^P)$.

VERIFIER(S_V, S'_V):
  for $i \in [k]$:
    $h_i \leftarrow$ VERIFIERITERATION(S_V) ▷ Protocol 3
  $i^* \leftarrow \mathrm{argmin}_{i\in[k]} L_{S'_V}(h_i)$
  output h_{i^*}

PROVER(S_P):
  loop forever:
    $q \leftarrow$ **receive** query from verifier
    $v \leftarrow \frac{1}{m_P}\sum_{i\in[m_P]} q\left(z_i^P\right)$
    send v to verifier

Protocol 2: A PAC verification protocol for statistical query algorithms.

Assumptions: As in Protocol 2.

VERIFIERITERATION(S_V):
  for $t \leftarrow 1, 2, \ldots$:
    simulate A until it sends a batch of queries or produces an output
    if A sends a batch of queries $\mathbf{q}_t$:
      if $t \geq b$:
        output reject and **terminate**
      $\mathbf{a}_t \leftarrow \mathsf{Atoms}(\sigma(\mathbf{q}_t))$
      send $\mathbf{a}_t$ to prover
      receive $\tilde{\mathbf{p}}_t$ from prover
      IDENTITYTEST($S_V, \mathbf{a}_t, \tilde{\mathbf{p}}_t, \tau$)
      $\tilde{\mathbf{v}}_t \leftarrow$ evaluations for $\mathbf{q}_t$ induced by $\tilde{\mathbf{p}}_t$
      send $\tilde{\mathbf{v}}_t$ to A
    else if A produces output h:
      return h

IDENTITYTEST($S_V, \mathbf{a}_t, \tilde{\mathbf{p}}_t, \tau$):
  for $j \in [m_V]$:
    $i_j \leftarrow i \in [|\mathbf{a}_t|]$ such that $a_t^i(z_j^V) = 1$
  execute the distribution identity tester of Theorem 15
    with sample $I = (i_1, \ldots, i_{m_V})$ to distinguish with
    probability at least $1 - \varepsilon\delta/4b$ between

$$\mathsf{TV}(\tilde{\mathbf{p}}_t, \mathbf{p}_t) \leq \frac{\tau}{2\sqrt{|\mathbf{a}_t|}}, \quad \text{and} \quad \tau \leq \mathsf{TV}(\tilde{\mathbf{p}}_t, \mathbf{p}_t)$$

    where $\mathbf{p}_t$ is the distribution that generated I
  if identity tester rejects:
    output reject and **terminate**

Protocol 3: A subroutine of Protocol 2.

Appendix C. Concentration of Measure

Theorem 30 **(Hoeffding, 1963)** *Let $a, b, \mu \in \mathbb{R}$ and $m \in \mathbb{N}$. Let $Z_1, \ldots, Z_m$ be a sequence of i.i.d. real-valued random variables and let $Z = \frac{1}{m}\sum_{i=1}^m Z_i$. Assume that $\mathbb{E}[Z] = \mu$, and for every $i \in [m]$, $\mathbb{P}[a \leq Z_i \leq b] = 1$. Then, for any $\varepsilon > 0$,*

$$\mathbb{P}[|Z - \mu| > \varepsilon] \leq 2\exp\left(\frac{-2m\varepsilon^2}{(b-a)^2}\right).$$