Efficient Learning with Arbitrary Covariate Shift

Adam Kalai ADUM@MICROSOFT.COM
Microsoft Research New England

Varun Kanade VARUNK@CS.OX.AC.UK
University of Oxford

Editors: Vitaly Feldman, Katrina Ligett and Sivan Sabato

Abstract

We give an efficient algorithm for learning a binary function in a given class C of bounded VC dimension, with training data distributed according to P and test data according to Q, where P and Q may be arbitrary distributions over X. This is the generic form of what is called *covariate shift*, which is impossible in general as arbitrary P and Q may not even overlap. However, recently guarantees were given in a model called PQ-learning (Goldwasser et al., 2020) where the learner has: (a) access to unlabeled test examples from Q (in addition to labeled samples from P, i.e., semi-supervised learning); and (b) the option to *reject* any example and abstain from classifying it (i.e., selective classification). The algorithm of Goldwasser et al. (2020) requires an (agnostic) noise-tolerant learner for C. The present work gives a polynomial-time PQ-learning algorithm, called *Slice-and-Dice*, that uses an oracle to a "reliable" learner for C, where reliable learning (Kalai et al., 2012) is a model of learning with one-sided noise. Furthermore, this reduction is optimal in the sense that we show the equivalence of reliable and PQ learning.

Keywords: PAC Learning, Covariate Shift

1. Introduction

Consider learning a binary function $f : X \to \{0, 1\}$ in a given class C of bounded VC dimension, with training data distributed according to P and test data according to Q, where P and Q may be arbitrary distributions over X. This form of what is often called learning with *Covariate Shift* (CvS) is extreme because P and Q may be arbitrary, whereas much work assumes bounded $Q(x)/P(x)$. In standard supervised learning, learning with arbitrary CvS is known to be impossible (e.g., Ben-David et al., 2010). In recent work, Goldwasser et al. (2020) show that it is possible using:

(a) Access to unlabeled test examples from Q (in addition to labeled samples from P). This is often called semi-supervised learning.

(b) The option to *reject* any example and abstain from classifying it. This is often called *selective classification* or classification with a reject option. Equivalently, one can think of a classifier that outputs 0, 1, or $\perp$, where $\perp$ indicates rejection.

(c) An efficient Empirical Risk Minimization oracle (ERM) which can find a classifier $c \in C$ of minimal error with respect to any training set. For C of bounded VC dimension, this is equivalent to "proper agnostic learning" as defined by Kearns et al. (1992), where agnostic learning is a model of learning with arbitrary label noise.

Goldwasser et al. (2020) call their model PQ-learning, and it is motivated by the numerous applications where the test distribution is different than the training distribution, for adversarial or natural reasons. Both (a) and (b) are provably necessary for the error and rejection requirements of PQ-learning, defined below. However, it is not clear what the *computational* requirements are—is PQ-learning as hard as agnostic learning or is it as easy as PAC learning?

We show PQ-learning is equivalent to learning with one-sided arbitrary noise, defined as *reliable learning* by Kalai et al. (2012). Crucial to this result, and perhaps the most interesting part of the paper, is the *Slice-and-Dice* (S&D) selective classification algorithm. S&D uses a reliable learner (rather than a full agnostic learner) to efficiently PQ learn. Conversely, we show that one cannot further reduce PQ-learning to a weaker oracle in that we also present a reduction from reliable learning to PQ-learning. Ignoring computation, the number of examples required for learning in all these models was known to be polynomially related to $d = \mathrm{VC}(C)$, the VC dimension of C.

Further, we give evidence that the difficulty of PQ-learning, and thus also reliable learning, lies somewhere in between that of PAC and agnostic learning, assuming the hardness of learning parity with noise and DNFs. In particular, we observe that parity functions are PQ-learnable. This suggests that PQ-learning is easier than agnostic learning since there is no known noise-tolerant parity learning algorithm, and in fact multiple cryptography systems rely on its hardness (see e.g., Pietrzak, 2012). We also observe that conjunctions, which are easily PAC learnable, are unlikely to be PQ-learnable, or at least that PQ-learning conjunctions would imply PAC-learning DNFs, a longstanding open PAC learning problem. Hence, there is an interesting computational complexity hierarchy in that PAC learning is easier than reliable-learning and PQ-learning, which in turn are easier than agnostic learning, assuming that parity is hard to agnostically learn and DNFs are hard to PAC-learn.

Previous work has shown such separations in related models. For example, Bshouty and Burroughs (2005) show that learning conjunctions in a model very similar to reliable learning implies PAC learning DNFs. Kanade and Thaler (2014) give an algorithm for reliably learning majorities over $\{0,1\}^d$ in time $2^{\tilde{O}(\sqrt{d})}$, whereas for this problem there are no agnostic learning algorithms known that run in time less than $2^{\Omega(d)}$. Although it doesn't neatly fit in the boolean function setting, Goel et al. (2017) showed that the class of "ReLUs" over $\{0,1\}^d$ is at least as hard to learn as learning $\omega(1)$-size parities with noise.

We now describe the learning models and our algorithms. S&D is intuitive and would be easy to implement in practice using off-the-shelf classifiers.

1.1. PQ-learning

Recall that the goal is to learn an unknown $f \in C$, where C is a given family of binary functions, with respect to arbitrary distributions P, Q over X. The learner is given examples from P labeled by an arbitrary $f \in C$, and unlabeled examples from Q. It outputs a *selective classifier* $h : X \to \{0, 1, \perp\}$, and we say that x is *rejected* if $h(x) = \perp$. An *error* is a misclassified example that is not rejected, i.e., $h(x) = 1 - f(x)$. Of course, one can guarantee 0 errors by simply rejecting everything, or 0 rejections by classifying everything as 0, but the challenge is to simultaneously achieve low error and rejection rates.

Now, one can consider the rejection and error rates with respect to P or Q. PQ-learning requires, with high probability, at most ϵ error rate with respect to Q, and at most ϵ rejection rate *with respect to* P. At first this may seem counter-intuitive, as one may care only about Q. Ideally, one would have liked a low rejection rate with respect to Q, but this is impossible in general since P and Q may be very different. However, if one is concerned with rejection rate with respect to Q, an ϵ P-rejection rate implies a Q-rejection rate of at most ϵ plus the statistical distance between P and Q. Thus, if $P = Q$ the rejection rate from Q is at most ϵ and the bound degrades naturally with the degree of overlap between P and Q. Also, as is standard, PQ learning requires the above for every $\epsilon > 0$ with a runtime (and thus also the number of labeled and unlabeled examples it uses) that is polynomial in $1/\epsilon$.

1.2. Reliable learning

Reliable learning, as defined by Kalai et al. (2012), is a model of learning with one-sided agnostic noise. Reliable learning is motivated by applications where false positives (or false negatives) are to be avoided at all cost. The reliable model applies to a standard agnostic setting (supervised learning with $P = Q$, no unlabeled test examples, no reject option, and arbitrary f, even $f \notin C$). For the moment, suppose that C is closed under complements, meaning that for any classifier $c \in C$, $1-c$ is also in C. Then reliably learning C means finding a classifier $h : X \to \{0, 1\}$ with ϵ false positive rate and which has an error rate at most $\mathrm{opt} + \epsilon$ where opt is the error rate of the best classifier with 0-false positive rate; assuming the constant 0 and 1 functions are in C such a classifier always exists. (An analogous notion for false negatives is equivalent if C is closed under complements.) Reliable learning has been applied in various works as described in the related work section below.

In practice, it is straightforward to implement a reliable learner using any standard classifier by a variety of means: one can heavily up-weight the negative examples, subsample the positive examples, use a cost-sensitive classification algorithm with high weight on false positive errors, or simply apply a positive label only on the examples that a classifier is most confident on for classifiers that also provide a confidence signal such as a margin. Thus there are a variety of ways to implement Algorithm 1 in practice. Formally, Kalai et al. (2012) prove that reliable learning is no harder than agnostic learning.

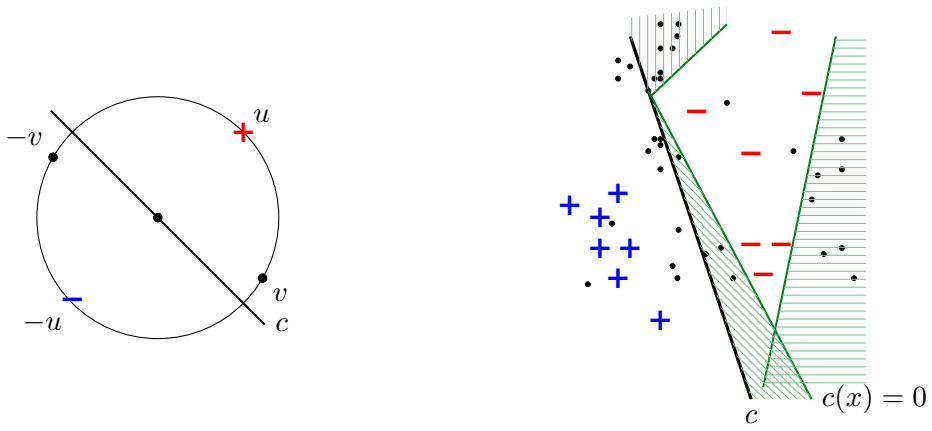
1.3. The Slice-and-Dice (S&D) Algorithm

The key question in selective classification and PQ learning is what to reject. Like previous selective classification algorithms, S&D first trains a classifier c on the labeled training data from P using, say, a PAC-learner. Like many such algorithms, S&D outputs a selective classifier h such that $h(x) \in \{c(x), \perp\}$, i.e., its classifications agree with c except that it rejects some examples.

To determine what to reject, S&D "slices" the space X into two parts: where $c(x) = 0$ versus $c(x) = 1$. It then rejects examples from each of these parts separately by repeated "dicing": distinguishing examples that clearly come from Q versus those that may come from P, on the respective part. To dice on $c(x) = 0$, it creates an artificial datasets of examples where $c(x) = 0$ consisting of both P-examples, labeled 0 and Q-examples, labeled 1. It then trains an sequence of positive reliable learners to distinguish P from Q, and h rejects examples that are classified as 1 by any such distinguisher. Now, not all examples from Q can be clearly distinguished from P (e.g., where say $Q(x) \leq 2P(x)$) but it turns out that not all examples need to be distinguished. It suffices to reject examples from Q that are clearly distinguishable from P by classifiers from C, but one must take care not to reject examples that are in fact from P (few false positives, hence positive reliable learning). A similar approach is used on the $c(x) = 1$ part.

The idea of rejecting examples by training a distinguisher to distinguish examples from Q versus P is intuitive. Unfortunately, this approach does not directly work without slicing the space into parts where $c(x) = 0$ and $c(x) = 1$. This is illustrated by a trivial 4-point halfspace example in Fig. 1(a), where C is the class of homogeneous halfspaces $\mathrm{sgn}(w_1x_1 + w_2x_2)$ that pass through the origin in two dimensions; P is uniform over two symmetric unit vectors $\{u, -u\}$; and Q is uniform over $\{u, -u, v, -v\}$, the same two points plus two additional symmetric unit vectors. Clearly $\pm v$ must be rejected because their labels are not determined by those of u, while we must not reject $\pm u$, but no homogeneous halfspace can distinguish $\pm u$ from $\pm v$. However, once one slices the space into positive and negative components, v can be distinguished from u within the respective parts. Fig. 1(b) illustrates S&D for halfspaces more generally.

In contrast, the Rejectron algorithm of Goldwasser et al. (2020) finds a sequence of candidate alternative classifiers that agree with the training data but disagrees with the test data and reject the disagreement regions. While this approach is also intuitive, it requires the full power of ERM (i.e., agnostic learning) rather than reliable learning and thus may require greater resources.



(a) (b)

Figure 1: (a) When learning homogeneous halfspaces (passing through the origin), there may be no halfspace that separates training examples $\pm u$ from test examples $\pm v$ even though the test examples must be rejected because the labels of $\pm u$ give no information about the labels of $\pm v$. Rejecting all points is not possible as we must not reject the points $\pm u$. (b) An illustration of S&D for general halfspaces that do not necessarily pass through the origin. When focusing on examples with $c(x) = 0$, we iteratively find classifiers in C that reliably separate the problematic $x \sim Q$ (which we will label as positive) from $x \sim P$ (which retain their labels and will be mostly negative) and reject the green-striped regions. By only focusing on the $c(x) = 0$ part we avoid having to find classifiers that also correctly classify the true positive examples.

Our solution is simple. Like the "Rejectron" algorithm of Goldwasser et al. (2020), we first train a classifier h on the labeled training data using, say, a PAC-learner. Then we separately focus on examples satisfying $h(x) = 1$ and $h(x) = 0$. Among those examples with $h(x) = 1$, we show that there must exist $c \in C$ that distinguishes at least some of the problematic $x \sim Q$ from $x \sim P$. And finding such a classifier amounts to a reliable classification problem, i.e., learning with one-sided noise.

1.4. Related work

This work is intimately related to that of Goldwasser et al. (2020) and Kalai et al. (2012). Goldwasser et al. (2020) introduces both PQ-learning and an adversarial transductive model of learning and exhibits a trade-off between rejection rates and accuracy that we do not study here. Reliable learning has also been studied within the learning theory community for a variety of applications (see e.g., Kanade and Thaler, 2014; Goel et al., 2017; Durgin and Juba, 2019). Several other models of learning are similar to reliable learning including models of Pitt and Valiant (1988); Juba (2016). In a related model, Bshouty and Burroughs (2005) show that learning conjunctions would imply learning DNF.

In supervised learning with CvS, a body of related work within the learning theory community studies learning with CvS assuming that $Q(x) \leq M \cdot P(x)$ for all $x \in X$ and some constant $M > 1$ (e.g., Huang et al., 2007; Ben-David and Urner, 2012). And without such an assumption supervised learning has been shown to be impossible (e.g., Ben-David et al., 2010).

A separate body of work studies selective classification (Rivest and Sloan, 1988; Li et al., 2011; Sayedi et al., 2010). Some of that work, particularly the work in online algorithms with a reject option, were targeted at non-stationary sequences of examples x. Even intervals are impossible to learn in online models, and in a supervised iid model Kivinen (1990) showed that exponentially many examples are required to learning rectangles under uniform distributions (as cited by Hopkins et al. (2019); Goldwasser et al. (2020)). Part of the challenge is that most definitions also require few test rejections, unlike PQ-learning's requirement of few rejections with respect to P. Other work, including the work of Kalai et al. (2012), makes the standard assumption that $P = Q$ and instead uses rejections to handle uncertain regions due to agnostic noise.

Finally, semi-supervised learning has been studied extensively in learning theory, again generally where $P = Q$. However, in the worst case the addition of unlabeled examples does not seem to provide significantly better guarantees than supervised learning (e.g., Ben-David et al., 2008).

2. Preliminaries and Background

Let X be the input space representing unlabeled examples and $Y = \{0, 1\}$ the target labels. Let Y^X denote the set of functions from $X \to Y$. For $f, h \in Y^X$, and a probability measure P over X, we denote the error of a hypothesis h with respect to a ground truth classifier f by

$$\mathrm{err}_P(h; f) = \mathbb{P}_{x \sim P}\left[h(x) \neq f(x)\right].$$

A *selective classifier* is a function $h : X \to \{0, 1, \perp\}$. For such a classifier $\mathrm{err}_P(h; f)$ is defined as,

$$\mathrm{err}_P(h; f) = \mathbb{P}_{x \sim P}\left[h(x) \neq f(x) \wedge h(x) \neq \perp\right].$$

We can also define the *rejection* rate of h with respect to a distribution P as

$$\mathrm{rej}_P(h) = \mathbb{P}_{x \sim P}\left[h(x) = \perp\right].$$

As observed by Goldwasser et al. (2020), the rejection rate with respect to P can be used to bound the rejection rate with respect to Q in multiple ways, including the following:

$$\mathrm{rej}_Q(h) = \mathbb{P}_{x\sim Q}\left[h(x) = \perp\right] \leq \mathrm{rej}_P(h) + \|P - Q\|_{TV},$$

where $\|P-Q\|_{TV}$ is the total variation distance (also called statistical distance) between P and Q.

We also denote the false positive rate of h with respect to a distribution D over $X \times \{0,1\}$ by

$$\mathrm{false}^+_D(h) = \mathbb{P}_{(x,y)\sim D}\left[h(x) = 1 \wedge y = 0\right].$$

Similarly the false negative rate of h with respect to a distribution D over $X \times \{0,1\}$ is denoted by

$$\mathrm{false}^-_D(h) = \mathbb{P}_{(x,y)\sim P}\left[h(x) = 0 \wedge y = 1\right].$$

A concept class $C \subseteq Y^X$ is a collection of functions from $X \to Y$. For classifier c, we denote the complementary classifier by $\bar{c} = 1 - c(x)$ and we denote $\bar{C} = \{\bar{c} \mid c \in C\}$. Following Kalai et al. (2012), we assume that the constant 0 and 1 functions are both in C in order for their notions of positive and negative reliable learning to be well-defined.

A distribution P over X and a function $f : X \to [0,1]$ define a joint probability distribution $D_{P,f}$ over $X \times \{0,1\}$, such that the marginal distribution over X is P and $\mathbb{E}_{(x,y)\sim D_{P,f}}[y|x] = f(x)$. Using this notation, the error rate of a classifier (regular or selective) can be decomposed into false positives and false negatives:

$$\mathrm{err}_P(h, f) = \mathrm{false}^+_{D_{P,f}}(h) + \mathrm{false}^-_{D_{P,f}}(h). \tag{1}$$

For a distribution P over X and a labeling function $f : X \to [0,1]$, we denote by $\mathsf{EX}(P, f)$ an example oracle that when queried gives a random *labeled* example from $D_{P,f}$. For distribution D over $X \times \{0,1\}$, we similarly define oracle $\mathsf{EX}(D)$. And for a distribution P over X, we denote by $\mathsf{EX}(P)$ an example oracle that when queried gives a random *unlabeled* example drawn from the distribution P. By a slight abuse of notation, $\alpha\mathsf{EX}(P_1, f_1) + (1-\alpha)\mathsf{EX}(P_2, f_2)$ denotes an oracle that returns a labeled example (x, y) where with probability α, (x, y) is drawn from D_{P_1,f_1}, and with probability $1-\alpha$, (x, y) is drawn from D_{P_2,f_2}. Finally, for an event A, $P|_A$ denotes the probability distribution P conditioned on A.

2.1. PQ Learning

We define the notion of PQ-Learning introduced in the recent work by Goldwasser et al. (2020). The definition below is mathematically identical to theirs, but presented slightly differently using the oracles $\mathsf{EX}(P, f)$ and $\mathsf{EX}(Q)$.

Definition 1 (PQ Learning (Goldwasser et al., 2020)) *A concept class C over X is PQ-learnable, if there exists a learning algorithm L, such that for every pair of distributions (P, Q) over X, every target $f \in C$, for every $\epsilon > 0$ and for every $\delta > 0$, when given access to the labeled example oracle $\mathsf{EX}(P, f)$ and the unlabeled example oracle $\mathsf{EX}(Q)$, $L(\epsilon, \delta, \mathsf{EX}(P, f), \mathsf{EX}(Q))$ outputs a selective classifier $h : X \to \{0, 1, \perp\}$, that with probability at least $1 - \delta$ simultaneously satisfies $\mathrm{err}_Q(h; f) \leq \epsilon$ and $\mathrm{rej}_P(h) \leq \epsilon$. Furthermore, L must run in time polynomial in $1/\epsilon$ and $1/\delta$.*

2.2. Reliable Learning

We now recall the definition of full reliable agnostic learning, introduced by Kalai et al. (2012), which we refer to as *reliable learning* for brevity. Formally, this is defined in terms of their notions of positive and negative (agnostic) reliable learning, which capture robustness to one-sided noise. However, it is worth noting that the three definitions are all equivalent for concept classes $C = \bar{C}$ that are closed under complements.

Definition 2 (Positive Reliable Learning) *A concept class C over X is positive reliably learnable, if there exists a learning algorithm L, such that for every distribution D over $X \times \{0,1\}$, every $\epsilon > 0$ and $\delta > 0$, when given access to the labeled example oracle $\mathsf{EX}(D)$, $L(\epsilon, \delta, \mathsf{EX}(D))$ outputs a hypothesis $h : X \to \{0,1\}$, that with probability at least $1-\delta$ satisfies $\mathrm{false}_D^+(h) \leq \epsilon$ and $\mathrm{false}_D^-(h) \leq \mathrm{opt}_+ + \epsilon$, where,*

$$\mathrm{opt}_+ = \min_{c \in C:\ \mathrm{false}_D^+(c)=0} \mathrm{false}_D^-(c).$$

Furthermore, we require the running time of L to be polynomial in $1/\epsilon$ and $1/\delta$.

A class C is said to be *negative reliably learnable* if $\bar{C}$ is positive reliably learnable, which optimizes subject to a restriction on false negatives. As mentioned, it must be assumed that the constant 0 (1) function is in C in order for positive (negative) reliable learning to be well-defined in the case where $f = 0$ ($f = 1$). Thus, as in Kalai et al. (2012), we assume that both the constant 0 and 1 functions are in C. We now define our main notion, reliable learning; it is easy to see that it is equivalent to positive reliable learning if $C = \bar{C}$.

Definition 3 (Reliable Learning) *A concept class C over X is reliably learnable if C and $\bar{C}$ are both positive reliably learnable.*

We note that the above notion is equivalent to the notion of *fully reliable agnostic learning* from Kalai et al. (2012). To define that notion, they considered a learning algorithm that outputs a selective classifiers h which has both false positive and negative rates bounded by ϵ. A pair of classifiers (c_+, c_-), where both $c_+, c_- \in C$, satisfying $\mathrm{false}^+(c_+) = 0$ and $\mathrm{false}^-(c_-) = 0$, can be converted to a reliable selective classifier which has no false positive or negative errors: output $c_+(x)$ if $c_+(x) = c_-(x)$ and output $\perp$ otherwise. Furthermore, it is required that the probability $\mathbb{P}[h(x) = \perp] \leq \mathrm{opt}_\perp + \epsilon$, where $\mathrm{opt}_\perp$ is the probability of predicting $\perp$ for the best pair of classifiers (c_+, c_-) as defined previously. To see that reliable learning is equivalent to fully reliable agnostic learning, note that given both positive and negative reliable learners, it is straightforward to construct an h with $\mathbb{P}[h(x) = \perp] \leq \mathrm{opt}_\perp + 2\epsilon$. And conversely, given a fully reliable learner, it is straightforward to construct positive and negative learners by simply converting its $\perp$ predictions to 0 or 1, respectively.

3. Equivalence Results

In this section, we prove our main result showing the equivalence between reliable learning and PQ learning up to polynomial factors in the running time (and sample complexity). Theorem 5 states that any concept class C that is reliably learnable is also PQ-learnable. Theorem 6 states the converse.

Before presenting the formal proof which relies on a slightly delicate argument, we give a high-level idea of the proof. In the slice step, S&D first finds a classifier c that PAC-learns the target f under the distribution P. This can be done since, without noise, a positive reliable learner (or a negative reliable learner) is also a PAC learner. The final classifier h will never have opposite labels with c but it may reject some examples, i.e., $h(x) \in \{c(x), \perp\}$ for all $x \in X$. Now we separately consider the parts where $c = 0$ and $c = 1$. Since they are symmetric, we'll focus on the part when $c = 0$. Our DICE_0 algorithm will distinguish Q from P, or at least where necessary. To do so, we will construct an artificial dataset and call a reliable learner on it. The dataset will consist entirely of examples where $c(x) = 0$, and subject to that condition it will be an equal mixture of examples drawn from P and from Q. The P examples will be labeled by f, which means that most, though not necessarily all, will also have $f(x) = 0$. We will label the examples drawn from Q as 1. Then we will find a *positive reliable classifier* for the resulting distribution. This allows us to identify a region that is almost exclusively Q, which we decide to reject. We repeat this process

Algorithm 1: The DICE$_0$ algorithm which takes as input a classifier c with error $\leq \epsilon/2$ and a positive reliable learning algorithm. It rejects part of the $c(x) = 0$ region so as to ensure a small false negative rate with respect to Q and small reject rate with respect to P.

Inputs:

a. Accuracy parameter ϵ, Confidence parameter δ

b. Access to oracles $\mathsf{EX}(P, f)$, $\mathsf{EX}(Q)$

c. Classifier $c : X \to \{0, 1\}$

d. Positive reliable learner L

let $M = \frac{2}{\epsilon} \cdot \log \frac{1}{\epsilon}$

if $\{\mathbb{P}_{x\sim P}[c(x) = 0] < \epsilon\}$ **then**

  define $h(x) = \begin{cases} c(x) & \text{if } c(x) = 1 \\ \perp & \text{otherwise} \end{cases}$

  Return: h

end

define the event $\mathcal{E}_1 = \{x \in X \mid c(x) = 0\}$

let $i = 1$

while $\{\mathbb{P}_{x\sim Q}[x \in \mathcal{E}_i] > \epsilon\}$ **do**

  $c_i = L\left(\frac{\epsilon}{2M}, \frac{\delta}{M}, \frac{1}{2}\mathsf{EX}(P|_{\mathbf{1}(c(x)=0)}, f) + \frac{1}{2}\mathsf{EX}(Q|_{\mathcal{E}_i}, 1)\right)$

  if $\{\mathbb{P}_{x\sim Q|_{\mathcal{E}_i}}[c_i(x) = 1] < \frac{\epsilon}{2}\}$ **then**

    break

  end

  $\mathcal{E}_{i+1} = \{x \in \mathcal{E}_i \mid c_i(x) = 0\} = \{x \in X \mid c(x) = c_1(x) = \ldots = c_i(x) = 0\}$

  $i = i + 1$

end

define $h(x) = \begin{cases} c(x) & \text{if } c(x) = 1 \text{ or } \bigwedge_i c_i(x) = 0 \\ \perp & \text{otherwise} \end{cases}$

Return: h

iteratively rejecting more and more of the space until we cannot find a nontrivial region to safely reject. The key observations are that: (a) the ground truth classifier f is always reliable, and thus we will continue to reject a non-trivial fraction of Q as long as there remain remains a non-negligible region under Q where f could output 1; and (b) each iteration rejects a region of small probability under P. In the end, we (mostly) only output 0 on examples under Q for which f is actually 0. The DICE_1 algorithm is completely analogous and runs on $c(x) = 1$ using negative reliable learning.

Lemma 4 *For any $\epsilon, \delta > 0$, distributions P, Q over X, $f \in C$, and $c : X \to \{0, 1\}$, with probability $\geq 1-\delta$, Algorithm 1 (DICE_0) returns a classifier h with $h(x) \in \{\perp, c(x)\}$ for all $x \in X$, $\text{rej}_P(x) \leq \epsilon + \text{err}_P(c; f)$, and $\text{false}^-_{D_{Q,f}}(h) \leq \epsilon$. The algorithm runs in expected time polynomial in $1/\epsilon, 1/\delta$ assuming access to an exact probability computation oracle required in Lines 2, 7 and 9.*

In other words, DICE_0 guarantees a low false positive rate by rejecting examples. Completely analogously, one defines DICE_1 and argues that used with a negative reliable learner, can guarantee a low false negative rate by rejecting further examples. The two can be applied to guarantee a low total error (by Eq. (1)) with bounds on the rejection rate with respect to P. Before we prove the above lemma, we show how to use it to prove the main theorem. The access to an exact probability computation oracle is not necessary, and estimates of the require probabilities, which can be obtained by sampling, are sufficient after minor adjustments to the constant factors, as explained in the proof of Theorem 5.

Theorem 5 *If a concept class C is reliably learnable, then C is PQ-learnable.*

Proof We first find c with $\text{err}_P(c; f) \leq \epsilon/4$ by running a positive (or negative) reliable learner with parameters $\epsilon/8, \delta/4$ on ordinary labeled examples $\mathsf{EX}(P, f)$. With probability $\geq 1 - \delta/4$ its output c has both false positive and false negative rates of $\epsilon/8$ (since $\text{opt}_+ = 0$). By the error decomposition in Eq. (1), this means it has error at most $\epsilon/4$.

Next, apply Algorithm 1 (DICE_0) to c with parameters $\epsilon/4$ and $\delta/4$. By Lemma 4, its output h_+ guarantees at most $\epsilon/4 + \epsilon/4 = \epsilon/2$ false negative rate with respect to Q and a rejection rate of at most $\epsilon/4$ with respect to P (with probability $\geq 1 - \delta/4$). Then run the symmetric version of Algorithm 1 (DICE_1) with a negative reliable learner to receive h_- with false positive rate at most $\epsilon/2$ with respect to Q, and a rejection rate of at most $\epsilon/4$ with respect to P (with probability $\geq 1 - \delta/4$). Finally, reject points that are rejected by either classifier, i.e., outputting the classifier:

$$h(x) = \begin{cases} h_+(x) & \text{if } h_+(x) = h_-(x) \\ \perp & \text{otherwise.} \end{cases}$$

By the union bound, it is easy to see that the rejection rate with respect to P would be at most ϵ. Further, by Eq. (1), the error rate with respect to Q would be at most ϵ, as required.

Unfortunately, Algorithm 1 is a hypothetical algorithm since it requires *exact* probability calculations in Lines 2, 7 and 9. In reality, the probabilities can only be estimated to a high accuracy with high probability. Using standard tedious arguments, one can design an algorithm without an exact probability oracle that enjoys the same guarantees of Lemma 4. It would involve a change of constants, including running the positive reliable learner with parameters that are a constant factor smaller, and straightforward applications of the Chernoff-Hoeffding bound and the union bound.

A second issue with Algorithm 1 is that it only runs in expected polynomial runtime, i.e. a "Las Vegas" algorithm. However, a standard timeout approach can be used to convert any Las Vegas algorithm into one that certainly runs in polynomial time (i.e. a "Monte Carlo" algorithm) and produces an identical result with probability $\geq 1 - \delta/4$. ■

We now prove Lemma 4.

Proof [of Lemma 4] First note that the lemma holds trivially if the condition in Line 2 holds, as rejecting all negative examples will necessarily lead to a false negative rate of 0 and a rejection rate of at most ϵ under P. So henceforth consider the case in which $\Pr_{x\sim P}[c(x) = 0] \geq \epsilon$.

For $i \geq 0$, let $q_i = \mathbb{P}_{x\sim Q}[x \in \mathcal{E}_i]$. Then, provided the condition in Line 9 does not cause the loop on Line 7 to break, we have that $q_i \leq q_{i-1} \cdot (1 - \epsilon/2)$. It will also terminate if $q_i \leq \epsilon$. Let j be the final value of i when the loop terminates. Thus,

$$\epsilon \leq (1 - \epsilon/2)^j \leq e^{-j\epsilon/2},$$

or equivalently $j \leq (2/\epsilon)\log(1/\epsilon) = M$.

Therefore, with probability $\geq 1 - j(\delta/M) \geq 1 - \delta$, all calls to the positive reliable learners succeed. Let us assume this is the happens.

We first bound the rejection rate, then the false negative rate, and finally analyze the runtime. Let D_i denote the distribution over $X \times \{0, 1\}$ produced by the example oracle used to learn c_i. Since the reliable learner succeeded, for each i we have,

$$\begin{aligned}
\frac{\epsilon}{2M} &\geq \text{false}^+_{D_i}(c_i) \\
&= \frac{1}{2}\Pr_{x\sim P}[c_i(x) = 1 \wedge f(x) = 0 \mid c(x) = 0] + \frac{1}{2}\Pr_{x\sim Q}[c_i(x) = 1 \mid \mathcal{E}_i] \\
&\geq \frac{1}{2}\Pr_{x\sim P}[c_i(x) = 1 \wedge f(x) = c(x) = 0] + 0 \\
\frac{\epsilon}{M} &\geq \Pr_{x\sim P}[c_i(x) = 1 \wedge f(x) = c(x) = 0]
\end{aligned}$$

In the above, we have used $\Pr[A \mid B] \geq \Pr[A \wedge B]$ by Bayes rule. This implies,

$$\begin{aligned}
\mathbb{P}_{x\sim P}[f(x) = 0 \wedge h(x) = \perp] &= \mathbb{P}_{x\sim P}[f(x) = c(x) = 0 \wedge \exists i c_i(x) = 1] \\
&\leq \sum_i \mathbb{P}_{x\sim P}[f(x) = c(x) = 0 \wedge c_i(x) = 1] \\
&\leq j\frac{\epsilon}{M} \leq M\frac{\epsilon}{M} = \epsilon.
\end{aligned}$$

Thus, as promised for the rejection bound,

$$\begin{aligned}
\mathbb{P}_{x\sim P}[h(x) = \perp] &= \mathbb{P}_{x\sim P}[f(x) = 0 \wedge h(x) = \perp] + \mathbb{P}_{x\sim P}[f(x) = 1 \wedge h(x) = \perp] \\
&\leq \epsilon + \mathbb{P}_{x\sim P}[f(x) = 1 \wedge c(x) = 0] \\
&\leq \epsilon + \text{err}_P(c; f).
\end{aligned}$$

We next bound the false negative rate, assuming that none of the positive reliable learners failed. Note that $h(x) = 0$ iff $x \in \mathcal{E}_j$. This implies:

$$\text{false}^-_{D_{Q,f}}(h) = \mathbb{P}_{x\sim Q}[f(x) = 1 \wedge h(x) = 0] = \mathbb{P}_{x\sim Q}[f(x) = 1 \wedge x \in \mathcal{E}_j]$$

Thus to bound the false negative rate, we must show,

$$\mathbb{P}_{x\sim Q}[f(x) = 1 \wedge x \in \mathcal{E}_j] \leq \epsilon \tag{2}$$

To do this, first note that if the while loop terminates because $\mathbb{P}_{x\sim Q}[x \in \mathcal{E}_j] \leq \epsilon$, then Eq. (2) holds trivially. So, suppose the loop termination is caused by the if statement on Line 9.

Now, by assumption $f \in C$ and by definition of D_j, $\text{false}^+_{D_j}(f) = 0$ and

$$\text{false}^-_{D_j}(f) = \frac{1}{2}\mathbb{P}_{x\sim Q|\mathcal{E}_j}[f(x) = 0]$$

On the other hand, we have $\text{false}^-_{D_j}(c_j) \geq \frac{1}{2}\mathbb{P}_{x\sim Q|\mathcal{E}_j}[c_j(x)=0]$ and also by the guarantee of the learning algorithm L, $\text{false}^-_{D_j}(c_j) \leq \text{false}^-_{D_j}(f) + \epsilon/(2M)$. Combining gives,

$$\frac{1}{2}\mathbb{P}_{x\sim Q|\mathcal{E}_j}[c_j(x)=0] \leq \text{false}^-_{D_j}(c_j) \leq \frac{1}{2}\mathbb{P}_{x\sim Q|\mathcal{E}_j}[f(x)=0] + \frac{\epsilon}{2M}$$

Rearranging and using the fact that $\Pr[f(x)=1] = 1 - \Pr[f(x)=0]$ gives,

$$\begin{aligned}\mathbb{P}_{x\sim Q|\mathcal{E}_j}[f(x)=1] &\leq \mathbb{P}_{x\sim Q|\mathcal{E}_j}[c_j(x)=1] + 2\frac{\epsilon}{2M} \\ &\leq \frac{\epsilon}{2} + \frac{\epsilon}{M} \leq \epsilon,\end{aligned}$$

where we have used the stopping criterion $\mathbb{P}_{x\sim Q|\mathcal{E}_j}[c_j(x)=1] \leq \epsilon/2$ from Line 9. Thus

$$\mathbb{P}_{x\sim Q}[f(x)=1 \wedge x \in \mathcal{E}_j] \leq \mathbb{P}_{x\sim Q}[f(x)=1 \mid \mathcal{E}_j] \leq \epsilon$$

The above is what we needed for Eq. (2) to bound the false negative rate. Since the total failure probability is at most δ, only the runtime analysis remains.

We have assumed that we have a unit-time oracle for exact probability calculations. To simulate samples from a mixed example oracle D_i, one flips a coin and chooses which oracle to sample from. To sample from a conditional oracle, one simply continues drawing samples until a sample satisfies the relevant event. The *expected* number of samples required is the reciprocal of the probability of the event. In this case, the expected number of samples required to get a sample from $\frac{1}{2}\mathsf{EX}(P|_{\mathbf{1}(c(x)=0)}, f) + \frac{1}{2}\mathsf{EX}(Q|_{\mathcal{E}_i}, 1)$ is at most,

$$\frac{1}{\min\left\{\mathbb{P}_{x\sim P}[c(x)=0], \mathbb{P}_{x\sim Q}[x\in\mathcal{E}_i]\right\}}.$$

But the algorithm tests in Lines 2 and 7 ensure that the above is $O(1/\epsilon)$. Since the positive reliable learner is called at most M times with parameters $\text{poly}(1/\epsilon, 1/\delta)$, it also runs in time $\text{poly}(1/\epsilon, 1/\delta)$. ■

Theorem 6 *If a concept class C is PQ-learnable, then C is reliably learnable.*

Proof We show that C is positive reliably learnable; the proof of negative reliable learnability is obtained *mutatis mutandis*. To reduce PQ-learning to reliable learning, given labeled examples from an arbitrary distribution D over $X \times \{0,1\}$, we must construct a noiseless distribution over labeled training examples and a distribution over unlabeled test examples. Our distribution over labeled training examples will simply be examples drawn from D whose labels are negative, which is consistent with the all 0 classifier which we have assumed is in C.

Define the distributions P, Q to be the marginal distributions over X conditioned on $y=0$ and $y=1$ respectively. Formally, for any measurable set $A \subseteq X$, they are defined by $P(A) = D(A\times\{0\})/D(X\times\{0\})$ and $Q(A) = D(A\times\{1\})/D(X\times\{1\})$. Sampling from P and Q can be performed by sampling from D and rejecting based on the value of y. The expected time to generate a sample from either distribution depends on the minimum class probability. To ensure this can be done efficiently, we suppose that $\mathbb{P}_{(x,y)\sim D}[y=0] > \epsilon/2$ and $\mathbb{P}_{(x,y)\sim D}[y=1] > \epsilon/2$. Otherwise a positive reliable learner is easily obtained by outputting a constant hypothesis. (A hypothesis with error at most ϵ must have both false positive and false negative rates of at most ϵ, but we use $\epsilon/2$ since we cannot compute probabilities exactly.)

Claim 4.2 in Goldwasser et al. (2020) shows that if you can PQ-learn then you can additionally guarantee $\text{err}_P(h; f) \leq \epsilon$. (This is done by simply running the PQ learner on distribution $Q' =$

$\frac{1}{2}P + \frac{1}{2}Q$.) Let us therefore assume that we have a PQ-learner L for C that guarantees $\mathrm{err}_Q(h; f)$, $\mathrm{err}_P(h; f)$, and $\mathrm{rej}_P(h)$ are all at most ϵ with probability $\geq 1 - \delta$.

Now, let $c^* \in C$ be a concept such that $\mathrm{false}^+_D(c^*) = 0$ and $\mathrm{false}^-_D(c^*) = \mathrm{opt}_+$. Note that unlike positive reliable learning, PQ-learning requires a target concept; we shall let c^* be the target concept as it is consistent with all examples under the probability distribution P. Thus, simply outputting a negative example drawn from D simulates the oracle $\mathsf{EX}(P, c^*)$. Likewise, the oracle $\mathsf{EX}(Q)$ is simulated by outputting the input part of a positively labeled example drawn from D. Let $h : X \to \{0, 1, \bot\}$ be the hypothesis returned by $L(\epsilon/2, \delta, \mathsf{EX}(P, c^*), \mathsf{EX}(Q))$. We use h to define a classifier $g : X \to \{0, 1\}$ as follows. Let $g(x) = 0$ if $h(x) = 0$, and $g(x) = 1$ if $h(x) \in \{\bot, 1\}$. Then we have the following:

$$\begin{aligned}
\mathrm{false}^+_D(g; D) &= \mathbb{P}_{(x,y)\sim D}\left[g(x) = 1 \wedge y = 0\right] \\
&\leq \mathbb{P}_{(x,y)\sim D}\left[g(x) = 1 \mid y = 0\right] \\
&= \mathbb{P}_{(x,y)\sim D}\left[h(x) = \bot \mid y = 0\right] + \mathbb{P}_{(x,y)\sim D}\left[h(x) = 1 \mid y = 0\right] \\
&\leq \mathrm{rej}_P(h) + \mathrm{err}_P(h; c^*) \leq \epsilon/2 + \epsilon/2 = \epsilon.
\end{aligned}$$

In the above we have used the rejection and err_P bound discussed above. Likewise,

$$\begin{aligned}
\mathrm{false}^-_D(g) &= \mathbb{P}_{(x,y)\sim D}\left[g(x) = 0 \wedge y = 1\right] \\
&= \mathbb{P}_{(x,y)\sim D}\left[h(x) = 0 \wedge y = 1\right] \\
&\leq \mathbb{P}_{(x,y)\sim D}\left[c^*(x) = 0 \wedge y = 1\right] + \mathbb{P}_{(x,y)\sim D}\left[h(x) \neq c^*(x) \wedge y = 1\right] \\
&\leq \mathbb{P}_{(x,y)\sim D}\left[c^*(x) = 0 \wedge y = 1\right] + \mathbb{P}_{(x,y)\sim D}\left[h(x) \neq c^*(x) \mid y = 1\right] \\
&= \mathrm{opt}_+ + \mathrm{err}_Q(h; c^*) \leq \mathrm{opt}_+ + \epsilon.
\end{aligned}$$

■

4. Separation Results

4.1. Algorithm for Learning Parities

In this section, we observe that a very simple algorithm can PQ-learn the class PARITIES of parity functions over $\{0, 1\}^d$. For $T \subseteq \{1, 2, \ldots, d\}$, the parity function $\oplus_T(x) := \oplus_{i \in T} x_i$ is 1 if an odd number of bits in S are 1. Note that the parity function corresponding to the empty set $\emptyset$ is the constant 0 function. If we also include the constant function 1, which is not expressible as a parity, in PARITIES, the reductions from the previous section also give positive and negative reliable learners for the class PARITIES.[1] We consider learnability of the family of classes $\langle \mathsf{PARITIES}_d \rangle_{d \geq 1}$, where $\mathsf{PARITIES}_d$ represents the parity functions over $\{0, 1\}^d$. Although the definitions in Section 2 omit this issue for readability, it is common in learning theory; for efficient learnability we require that the running time is bounded by a polynomial in d, $1/\epsilon$ and $1/\delta$.

Lemma 7 *The class* PARITIES *is PQ-learnable.*

The proof idea is that we may reject all examples not in the span of the training data—all other examples have a label that can be uniquely determined from the training data. The complete proof appears in Appendix A.

1. The inclusion of the constant 1 function is only required for *negative reliable learning*. Without it there is no guarantee that there is any concept in the class that has no false negative errors.

4.2. Hardness of Learning Conjunctions

The equivalence of PQ and reliable learning makes it easy to show that PQ learning is likely harder than PAC learning. In this section, we observe that positively reliably learning the class C of conjunctions is as hard as the problem of PAC learning DNF formulae, a problem that has remained open since the seminal paper of Valiant (1984) that introduced it. Learning DNF formulae is known to be at least as hard as learning juntas on $\log d$ variables, another notoriously hard problem (Blum, 1993), and has recently shown to be hard conditional on the hardness of refuting random k-SAT formulae (Daniely and Shalev-Shwartz, 2016). As in the previous section, we consider a family of concept classes C_d parametrized by size complexity d.

For $X = \{0,1\}^d$ and $S \subseteq \{1, 2, \ldots d\}$, the conjunction $\wedge_S$ is the function which is 1 if all the bits of S are 1. The family $\wedge_{\pm d}$ of *general conjunctions* includes conjunctions that may have literals or negations of literals. We first note that if one can positively reliably learn $\wedge_d$ then one can positively reliably learn $\wedge_{\pm d}$. This follows from a standard representation trick—one simply maps each example in $\{0,1\}^d$ to an example $x' \in \{0,1\}^{2d}$ by taking each $x \in \{0,1\}^d$ and concatenating x with the bits of x negated. In this representation, any general conjunction over x corresponds to a conjunction over x'. Although not directly relevant, we note that without noise, the family $\wedge_d$ of conjunctions on d variables is trivially learnable from positive examples alone (or online with at most d mistakes), and thus also negative reliably learnable.

The class of s-term DNF formulae over $\{0,1\}^d$ consists of boolean functions that can be represented as a disjunction of at most s terms, $\varphi = T_1 \vee T_2 \vee \cdots T_s$, where each term $T_i \in \wedge_{\pm d}$. It is easy to see that a positive reliable learner for conjunctions can be used to *weakly* learn s-term DNF formulae. If the labels under the distribution are not (almost) balanced, then either the constant 0 or 1 function is already a weak learner. Otherwise, each of the conjunctions T_i classifies all examples labeled negatively by φ as negative, and at least one of the T_i's classifies $\frac{1}{s}$ fraction of the examples labeled positively by φ correctly. Thus, a positive reliable learner for conjunctions yields a $\Omega(1/s)$ weak learner for s-term DNF. A standard boosting algorithm can be then used to convert this to obtain a PAC learning algorithm. We remark that observations along these lines have already been made in previous work (e.g. Bshouty and Burroughs (2005); Kalai et al. (2012)). The above discussion can be formalized in the form of the following lemma whose proof is omitted.

Lemma 8 *If conjunctions are positively reliably learnable, then polynomial-size* DNF *formulae are PAC learnable.*

Acknowledgments

Varun Kanade was supported in part by the Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

S. Ben-David, Tyler Lu, Teresa Luu, and D. Pál. Impossibility theorems for domain adaptation. In *AISTATS*, 2010.

Shai Ben-David and Ruth Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *International Conference on Algorithmic Learning Theory*, pages 139–153. Springer, 2012.

Shai Ben-David, Tyler Lu, and Dávid Pál. Does unlabeled data provably help? Worst-case analysis of the sample complexity of semi-supervised learning. In *COLT*, pages 33–44, 2008.

Avrim L. Blum. Relevant examples and relevant features: Thoughts from computational learning theory. In *In AAAI-94 Fall Symposium on 'Relevance', 1994.*, pages 302–311, 1993.

N. Bshouty and L. Burroughs. Maximizing agreements with one-sided error with applications to heuristic learning. *Machine Learning*, 59:99–123, 2005.

Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning DNF's. In *Conference on Learning Theory*, pages 815–830, 2016.

Alexander Durgin and Brendan Juba. Hardness of improper one-sided learning of conjunctions for all uniformly falsifiable CSPs. In *Algorithmic Learning Theory*, pages 369–382, 2019.

Surbhi Goel, Varun Kanade, Adam Klivans, and Justin Thaler. Reliably learning the ReLU in polynomial time. In *Conference on Learning Theory*, pages 1004–1042. PMLR, 2017.

Shafi Goldwasser, Adam Tauman Kalai, Yael Tauman Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020*, 2020. URL https://arxiv.org/abs/2007.05145.

Max Hopkins, Daniel M Kane, and Shachar Lovett. The power of comparisons for actively learning linear classifiers. *arXiv preprint arXiv:1907.03816*, 2019.

Jiayuan Huang, Arthur Gretton, Karsten Borgwardt, Bernhard Schölkopf, and Alex J Smola. Correcting sample selection bias by unlabeled data. In *Advances in neural information processing systems*, pages 601–608, 2007.

Brendan Juba. Learning abductive reasoning using random examples. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pages 999–1007, 2016.

Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012.

Varun Kanade and Justin Thaler. Distribution-independent reliable learning. In *Conference on Learning Theory*, pages 3–24, 2014.

Michael J. Kearns, Robert E. Schapire, Linda M. Sellie, and Lisa Hellerstein. Toward efficient agnostic learning. In *In Proceedings of the Fifth Annual ACM Workshop on Computational Learning Theory*, pages 341–352, 1992.

Jyrki Kivinen. Reliable and useful learning with uniform probability distributions. In *Proceedings of the First International Workshop on Algorithmic Learning Theory (ALT)*, pages 209–222, 1990.

Lihong Li, Michael L Littman, Thomas J Walsh, and Alexander L Strehl. Knows what it knows: a framework for self-aware learning. *Machine learning*, 82(3):399–443, 2011.

Krzysztof Pietrzak. Cryptography from learning parity with noise. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 99–114. Springer, 2012.

Leonard Pitt and Leslie G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4): 965–984, October 1988. ISSN 0004-5411. doi: 10.1145/48014.63140. URL https://doi.org/10.1145/48014.63140.

Ronald L Rivest and Robert H Sloan. Learning complicated concepts reliably and usefully. In *AAAI*, pages 635–640, 1988.

Amin Sayedi, Morteza Zadimoghaddam, and Avrim Blum. Trading off mistakes and don't-know predictions. In *Advances in Neural Information Processing Systems*, pages 2092–2100, 2010.

Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Appendix A. Proof from Section 4

The proof of Lemma 7 gives an algorithm for PQ Learning Parities.

Proof [of Lemma 7] Let f be the target parity distribution. Observe that the VC dimension of $\mathsf{PARITIES}$ is d and let $m = \mathrm{poly}(d, 1/\epsilon, 1/\delta)$ be large enough that so that for a sample S of size m drawn from the oracle $\mathsf{EX}(P, f)$ satisfies with probability at least $1 - \delta$, that for all $c \in \mathsf{PARITIES}$,

$$|\mathrm{err}_{\delta_S}(c; f) - \mathrm{err}_P(c; f)| \leq \epsilon/2,$$

where δ_S is the empirical distribution over the sample S.

Let $\hat{c} \in \mathsf{PARITIES}$ be any parity function for which $\mathrm{err}_{\delta_S}(\hat{c}; f) = 0$. Such a $\hat{c}$ must exist as $f \in \mathsf{PARITIES}$, and in fact can be found efficiently using Gaussian elimination. Let $S = \{(x_1, y_1), \ldots (x_m, y_m)\}$ denote the training dataset. Considering $\{0,1\}^d$ as a vector space over $\mathbf{GF}(2)$, let k denote the dimension of the vector space $V := \mathrm{span}\{x_1, \ldots, x_m\}$. If $k = d$, then $\hat{c}$ is uniquely determined and is equal to the target parity f. Otherwise, let $\{a_1, \ldots, a_{d-k}\} \subseteq \{0,1\}^d$ be a set of linearly independent vectors so that $\mathrm{span}\{x_1, \ldots, x_m, a_1, \ldots, a_{d-k}\} = \{0,1\}^d$. Assigning a label of either 0 or 1 to the "*datum*" a_i uniquely defines a parity that is consistent with $\hat{c}$ on S. Thus, there are 2^{d-k} possible choices of $\hat{c}$ that are consistent with S. Let us denote this set by $\mathsf{PARITIES}_S$. We use this fact to prove that $\mathbb{P}_{x\sim P}[x \notin V] \leq \epsilon$. Clearly, if V is d-dimensional, then this probability is 0. Otherwise any $x \notin V$ can be written as $x' + \tilde{x} + a_i$ for some $i \in \{1, \ldots, d-k\}$ and for $x' \in V$ and $\tilde{x} \in \mathrm{span}\{a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_{d-k}\}$. Now for a random parity $\tilde{c}$ drawn from $\mathsf{PARITIES}_S$, the probability that $f(x) \neq \tilde{c}(x)$ is exactly $\frac{1}{2}$. To see this, observe that a random parity can be chosen by first assigning a random label from $\{0,1\}$ to each a_j, $j \neq i$. Then the label of $\tilde{c}(x)$ can still be either 0 or 1 and is completely determined by the random choice made for a_i. This means that the label assigned will be different from $f(x)$ with probability exactly $\frac{1}{2}$. Thus, we have:

$$\mathop{\mathbb{E}}_{\tilde{c}\sim_U \mathsf{PARITIES}_S}[\mathrm{err}_P(\tilde{c}; f)] \geq \frac{1}{2}\mathbb{P}_{x\sim P}[x \notin V]$$

On the other hand, by our choice of m, with probability at least $1 - \delta$, we have that for each $\tilde{c} \in \mathsf{PARITIES}_S$, $\mathrm{err}_P(\tilde{c}; f) \leq \epsilon/2$. Hence, it must be the case that $\mathbb{P}_{x\sim P}[x \notin V] \leq \epsilon$.

Let $h : \{0,1\}^d \to \{0, 1, \perp\}$ be defined as follows: if $x \in V$, then $h(x) = \hat{c}(x)$, else $h(x) = \perp$. Clearly, for any $x \in V$, $\hat{c}(x) = f(x)$, hence $\mathrm{err}_Q(h; f) = 0$. On the other hand, $\mathrm{rej}_P(h) \leq \mathbb{P}_{x\sim P}[x \notin V] \leq \epsilon$. ■