Proceedings of Machine Learning Research vol 65:1–28, 2017

Predicting with Distributions

Michael Kearns MKEARNS@CIS.UPENN.EDU and **Zhiwei Steven Wu** WUZHIWEI@CIS.UPENN.EDU
University of Pennsylvania

Abstract

We consider a new learning model in which a joint distribution over vector pairs (x, y) is determined by an unknown function $c(x)$ that maps input vectors x not to individual outputs, but to entire *distributions* over output vectors y. Our main results take the form of rather general reductions from our model to algorithms for PAC learning the function class and the distribution class separately, and show that virtually every such combination yields an efficient algorithm in our model. Our methods include a randomized reduction to classification noise and an application of Le Cam's method to obtain robust learning algorithms.

Keywords: PAC learning, distribution learning, learning with classification noise.

1. Introduction

We consider a new variant of the *Probably Approximately Correct (PAC)* learning framework. In our model, a joint distribution over vector pairs (x, y) is determined by an unknown target function $c(x)$ that maps input vectors x not to individual outputs, but to entire *distributions* over output vectors y in some large space. This model generalizes settings such as learning with classification noise or errors, probablistic concepts (where y is a probabilistic but scalar function of x), multiclass learning (where y is a multi- or vector-valued but deterministic function of x), and settings in which the output space associated with a classification may be large and complex. It is an instance of a more general framework in which the distribution of multiple hidden variables — with unknown but parametric structural dependencies on observable inputs — determines the distribution of observable outputs. For the special case of a single binary hidden variable, we provide the first formal learning guarantees in a PAC framework.

As in the standard PAC model, we begin with an unknown binary function or concept c chosen from a known class $\mathcal{C}$,[1] whose inputs x are distributed according to an unknown and arbitrary distribution. Now, however, the value $c(x)$ determines which of two unknown probability distributions $P_{c(x)}$ govern the distribution of y, where P_0 and P_1 are chosen from a known class of distributions $\mathcal{P}$. Thus y is distributed according to a mixture model, but the mixture component is given by a hidden classifier c. The learner does not see explicit labels $c(x)$, but only the resulting (x, y) pairs. The goal is to learn a *hypothesis model* that consists of a hypothesis h that is a $\{0, 1\}$-valued function, and two probability distributions $\hat{P}_0$ and $\hat{P}_1$ from the class $\mathcal{P}$. Given any input x, the model will predict the vector y to be drawn from the distribution $\hat{P}_{h(x)}$ (and hence *predict with distribution* $\hat{P}_{h(x)}$). Our objective is to minimize the *conditional* Kullback-Leibler (KL) divergence $\mathbb{E}_x \left[\mathrm{KL}(P_{c(x)} || \hat{P}_{h(x)})\right]$, rather than simply the KL divergence to the mixture. We thus refer to our model as *Predicting with Distributions (PwD)*.

1. We leave the consideration of multi- or real-valued functions $c(x)$ to future work.

One of our primary motivations is *composition* and *reducibility* across different learning models — in this case, models for classification and models for distribution learning. Within the standard PAC (classification) model, there is a rich theory of reducibility between specific learning problems (Pitt and Warmuth, 1990; Kearns and Valiant, 1994), between classes of learning problems (Schapire, 1990; Kearns, 1998), as well as composition theorems allowing the creation of more complex learning algorithm from simpler ones (Kearns et al., 1994a). Less common are results allowing one to assemble algorithms with provable performance guarantees from constituents that are solving different *types* of learning problems. A natural starting point for such an investigation is with the standard PAC supervised learning model, and its distributional analogue (Kearns et al., 1994b), since these models are each already populated with a number of algorithms with strong theoretical guarantees.

Our main technical interest is thus in conditions permitting *computationally* efficient learning algorithms composed of extant classification and distribution learning algorithms. Informally, our results imply that for every concept class $\mathcal{C}$ known to be PAC learnable with classification noise (Angluin and Laird, 1987), and almost every class $\mathcal{P}$ known to be PAC learnable in the distributional sense of Kearns et al. (1994b), PwD problems given by $(\mathcal{C}, \mathcal{P})$ are learnable in our framework.

1.1. Our Results and Techniques

Our results take the form of reductions from our model to algorithms for PAC learning the concept class $\mathcal{C}$ and the distribution class $\mathcal{P}$ separately.[2] The primary conceptual step is in identifying the natural technical conditions that connect these two different classes of learning problems. The centerpiece in this "bridge" is the notion of a *distinguishing event* for two probability distributions $P_0, P_1 \in \mathcal{P}$, which is an event whose probability is "signficantly" (inverse polynomially) different under P_0 and P_1, provided these distributions are themselves sufficiently different.

Our first result shows that a distinguishing event can be used, via a particular randomized mapping, to turn the observed y into a noisy binary label for the unknown concept c. This will serve as a building block for us to combine efficient PAC learners from classification and distribution learning.

We then use distinguishing events to provide two different reductions of our model to PAC classification and distribution learning algorithms. In the "forward" reduction, we assume the distribution class $\mathcal{P}$ admits a small set of candidate distinguishing events. We show that such candidate events exist and can be efficiently constructed for the class of spherical Gaussians and product distributions over any discrete domain. By searching and verifying this set for such an event, we first PAC learn c from noisy examples, then use the resulting hypothesis to "separate" P_0 and P_1 for a distributional PAC algorithm for the class $\mathcal{P}$. This gives:

Theorem 1 (*Informal Statement, Forward Reduction*) *Suppose that the concept class $\mathcal{C}$ is PAC learnable under classification noise, and the distribution class $\mathcal{P}$ is PAC learnable and admits a polynomial-sized set of distinguishing events. Then the joint class $(\mathcal{C}, \mathcal{P})$ is PwD-learnable.*

In the "reverse" reduction, we instead first separate the distributions, then use their approximations to learn c. Here we need a stronger distribution-learning assumption, but no assumption on distinguishing events. More precisely, we assume that *mixtures* of two distributions from $\mathcal{P}$

2. Throughout the paper, all PAC learning algorithms (for both concept class $\mathcal{C}$ and distribution class $\mathcal{P}$) in our reduction runs in polynomial time, since we are primarily concerned with computational efficiency (as opposed to sample complexity).

(which is exactly what the unconditioned y is) are PAC learnable. Once we have identified the (approximate) mixture components, we show they can be used to explicitly *construct* a specialized distinguishing event, which in turn lets us create a noisy label for c. This leads our result in the reverse reduction:

Theorem 2 (*Informal Statement, Reverse Reduction)* *Suppose that the concept class $\mathcal{C}$ is PAC learnable under classification noise, and any mixture of two distributions from $\mathcal{P}$ is PAC learnable. Then the joint class $(\mathcal{C}, \mathcal{P})$ is PwD-learnable.*

In both reductions, we make central use of Le Cam's method to show that any PAC concept or distribution learning algorithm must have a certain "robustness" to corrupted data. Thus in both the forward and reverse directions, by controlling the accuracy of the model learned in the first step, we ensure the second step of learning will succeed.

Since practically every $\mathcal{C}$ known to be PAC learnable can also be learned with classification noise (either directly or via the statistical query framework (Kearns, 1998), with parity-based constructions being the only known exceptions), and the distribution classes $\mathcal{P}$ known to be PAC learnable have small sets of distinguishing events (such as product distributions), and/or have mixture learning algorithms (such as Gaussians), our results yield efficient PwD algorithms for almost all combinations of PAC classification and distribution learning algorithms known to date.

1.2. Related Works

At the highest level, our model falls under the framework of Haussler (1992), which gives a decision-theoretic treatment of PAC-style learning (Valiant, 1984) for very general loss functions; our model can be viewed as a special case in which the loss function is conditional log-loss given the value of a classifier. Whereas Haussler (1992) is primarily concerned with sample complexity, our focus here is on computational complexity and composition of learning models.

At a more technical level, our results nicely connect two well-studied models under the PAC learning literature. First, our work is related to the results in PAC learning under classification noise (Angluin and Laird, 1987; Decatur, 1997; Kearns, 1998), and makes use of a result by Ralaivola et al. (2006) that established the equivalence of learning under (standard) classification noise (CN) and under class-conditional classification noise (CCCN). Our work also relies on the PAC model for distribution learning (Kearns et al., 1994b), including a long line of works on learning mixtures of distributions (see e.g. Dasgupta (1999); Arora and Kannan (2001); Vempala and Wang (2004); Feldman et al. (2008)). Our new model of PwD learning, in particular, can be viewed as a *composition* of these two models.

Our model is also technically related to the one of co-training (Blum and Mitchell, 1998) in that the input x and the output y give two different views on the data, and they are conditionally independent given the unknown label $z = c(x)$, which is also a crucial assumption for co-training (as well as various other latent variable models for inference and learning). However, our model is also fundamentally different from co-training in two ways. First, in our model, there is not a natural target Boolean function that maps y to the label z. For example, any outcome y can be generated from both distributions P_0 and P_1. In other words, just using y is not sufficient for identifying the label z. Second, our learning goal is to predict what distribution the outcome y is drawn from given the input x, as opposed to predicting the unknown label z.

2. Preliminaries

2.1. Model: PwD-Learning

Let $\mathcal{X}$ denote the space of all possible *contexts*, and $\mathcal{Y}$ denote the space of all possible *outcomes*. We assume that all contexts $x \in \mathcal{X}$ are of some common length n, and all outcomes $y \in \mathcal{Y}$ are of some common length k. Here the lengths are typically measured by the dimension; the most common examples for $\mathcal{X}$ are the boolean hypercube $\{0,1\}^n$ and subsets of $\mathbb{R}^n$ ($\{0,1\}^k$ and $\mathbb{R}^k$ for $\mathcal{Y}$).

Let $\mathcal{C}$ be a class of $\{0,1\}$-valued functions (also called *concepts*) over the context space $\mathcal{X}$, and $\mathcal{P}$ be a class of probability distributions over the outcome space $\mathcal{Y}$. We assume an *underlying distribution* $\mathcal{D}$ over $\mathcal{X}$, a *target concept* $c \in \mathcal{C}$, and *target distributions* P_0 and P_1 in $\mathcal{P}$. Together, we will call the tuple (c, P_0, P_1) the *target model*.

Given any target model (c, P_0, P_1) and underlying distribution $\mathcal{D}$, our learning algorithm is then given sample access to the following generative example oracle $\text{Gen}(\mathcal{D}, c, P_0, P_1)$ (or simply Gen). On each call, the oracle does the following (see Figure 1 for an illustration):

1. Draws a context x randomly according to $\mathcal{D}$;
2. Evaluates the concept c on x, and draws an outcome y randomly from $P_{c(x)}$;
3. Returns the context-outcome pair (x, y).

A *hypothesis model* is a triple $T = (h, \hat{P}_0, \hat{P}_1)$ that consists of a *hypothesis* $h \in \mathcal{C}$ and two *hypothesis distributions* $\hat{P}_0$ and $\hat{P}_1 \in \mathcal{P}$. Given any context x, the hypothesis model predicts the outcome y to be drawn from the distribution $\hat{P}_{h(x)}$ (or simply *predicts with distribution* $\hat{P}_{h(x)}$). The goal of our learning algorithm is to output a hypothesis model with high accuracy with respect to the target model, and the error of any model T is defined as

$$err(T) = \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \left[\mathrm{KL}(P_{c(x)} || \hat{P}_{h(x)}) \right]$$

where KL denotes Kullback-Leibler divergence (KL divergence).

Our model of *Predicting with Distributions learning (PwD-learning)* is thus defined as follows.

Definition 3 (PwD-Learnable) *Let $\mathcal{C}$ be a concept class over $\mathcal{X}$, and $\mathcal{P}$ be a class of distributions over $\mathcal{Y}$. We say that the joint class $(\mathcal{C}, \mathcal{P})$ is* PwD-learnable *if there exists an algorithm $\mathcal{L}$ such that for any target concept $c \in \mathcal{C}$, any distribution $\mathcal{D}$ over $\mathcal{X}$, and target distributions $P_0, P_1 \in \mathcal{P}$ over $\mathcal{Y}$, and for any $\varepsilon > 0$ and $0 < \delta \leq 1$, the following holds: if $\mathcal{L}$ is given inputs ε, δ as inputs and sample access from $\text{Gen}(\mathcal{D}, c, P_0, P_1)$, then $\mathcal{L}$ will halt in time bounded by $\text{poly}(1/\varepsilon, 1/\delta, n, k)$ and output a triple $T = (h, \hat{P}_0, \hat{P}_1) \in \mathcal{C} \times \mathcal{P} \times \mathcal{P}$ that with probability at least $1 - \delta$ satisfies $err(T) \leq \varepsilon$.*

Observe that the unconditional distribution over y is a mixture of the target distributions P_0 and P_1. In our model, it is not enough to learn the mixture distribution (which is a standard problem in learning mixtures of distributions). Our learning objective is to minimize the expected *conditional KL divergence*, which is more demanding and in general requires a good approximation to the target concept c over $\mathcal{X}$.

Also note that we have stated the definition for the "proper" learning case in which the hypothesis models lie in the target classes $\mathcal{C}$ and $\mathcal{P}$. However, all of our results hold for the more general case in which they lie in potentially richer classes $\mathcal{C}'$ and $\mathcal{P}'$.

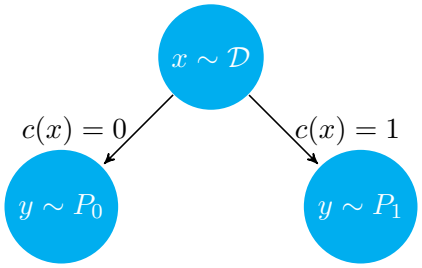

Figure 1: The generative model Gen: (1) first draw a context x from the underlying distribution $\mathcal{D}$, (2) then evaluate the concept c on x and (3) draw the outcome y from distribution $P_{c(x)}$.

2.2. Related Learning Models

We now discuss two learning models related to our setting (see the appendix for formal definitions).

CN Learning We first introduce PAC learning under *classification noise (CN)* (Angluin and Laird, 1987). For any *noise rate* $0 \leq \eta < 1/2$, consider the example oracle $EX^{\eta}_{\mathrm{CN}}(c, \mathcal{D})$ that on each call draws an example $(x, c(x))$ randomly according to $\mathcal{D}$, then with probability $1 - \eta$ returns the uncorrupted example $(x, c(x))$, and with probability η returns the erroneous example $(x, \neg c(x))$. The concept class $\mathcal{C}$ is *CN learnable* if there exists a polynomial-time algorithm that given sample access to EX^{η}_{CN} finds a hypothesis $h \in \mathcal{C}$ that approximately minimizes the classification error: $err(h) = \Pr_{x \sim \mathcal{D}}[c(x) \neq h(x)]$.

CCCN Learning In a more general noise model called *Class-Conditional Classification Noise (CCCN)* proposed by Ralaivola et al. (2006), the example oracle $EX^{\eta}_{\mathrm{CCCN}}$ has class-dependent noise rates — that is, the noise rate η_0 for the negative examples ($c(x) = 0$) and the noise rate η_1 for the positive examples ($c(x) = 1$) may be different, and both below 1/2. Moreover, Ralaivola et al. (2006) show that any class that is learnable under CN is also learnable under CCCN. (See the appendix for a formal statement).

Distribution Learning We also make use of results from for *PAC learning probability distributions* (Kearns et al., 1994b). A distribution class $\mathcal{P}$ is efficiently learnable if there exists a polynomial-time algorithm that, given sample access to an unknown target distribution P, outputs an accurate distribution $\hat{P}$ such that $\mathrm{KL}(P||\hat{P}) \leq \varepsilon$ for some target accuracy ε. For any distribution $P \in \mathcal{P}$ and any point $y \in \mathcal{Y}$, we assume that we can evaluate the probability (density) of y assigned by P (referred to as learning with an *evaluator* in Kearns et al. (1994b); see the appendix for the formal definition). We will write $P(y)$ to denote the probability (or density) of point y, and write $P(E)$ to denote $\Pr_{y \sim P}[y \in E]$ for any measurable set $E \subset \mathcal{Y}$.

To simplify our analysis, for the remainder of the paper we will make the following assumption on the class $\mathcal{P}$ to ensure that the log-likelihood loss (or log-loss) is bounded in the domain $\mathcal{Y}$. While this condition may not hold for some natural classes of distributions (e.g. Gaussians), it can be obtained using standard procedures (for instance, by truncating, or mixing with a small amount of the uniform distribution; see Feldman et al. (2006) for an example).

Assumption 4 (Boundedness Assumption) *There exists a quantity M that is upper bounded by* $\mathrm{poly}(k)$ *such that for any distribution $P \in \mathcal{P}$ and any point $y \in \mathcal{Y}$, we have* $\log(1/P(y)) \leq M$.

3. CN Learning with Identified Distinguishing Events

In this section, we will introduce a central concept to our framework—*distinguishing events*. Informally, an event $E \subset \mathcal{Y}$ is distinguishing for distributions P_0 and P_1 if it occurs with different probabilities under the measures of P_0 and P_1. As a consequence, these events are informative about target concept c that determines which distribution the outcome y is drawn from. We will rely on such events to create a CCCN learning instance for the target concept c. Thus, whenever the class $\mathcal{C}$ is learnable under CN (and hence learnable under CCCN by Ralaivola et al. (2006)), we can learn the target concept c under the PwD model using a distinguishing event.

Definition 5 (Distinguishing Event) *Let P and Q be distributions over the outcome space $\mathcal{Y}$, and let $\xi > 0$. An event $E \subseteq \mathcal{Y}$ is ξ-distinguishing for distributions P and Q if $|P(E) - Q(E)| \geq \xi$. We will call ξ the* separation parameter *for such an event.*

We will now show that the knowledge of a distinguishing event between P_0 and P_1 allows us to simulate an example oracle $EX^{\eta}_{\mathrm{CCCN}}$, and therefore we can learn the concept c with a CCCN learner. The main technical problem here is to assign noisy labels based on the distinguishing event so that noise rates η_0 and η_1 of the oracle are strictly less than $1/2$.

Our solution is to construct a *randomized mapping* from the event to the labels.[3] Let us first introduce some parameters. Let $E \subseteq \mathcal{Y}$ be a ξ-distinguishing event for the distributions P_0 and P_1 for some $\xi \in (0, 1]$. We will write $p = P_0(E)$ and $q = P_1(E)$. Consider the following algorithm $\mathbf{Lab}(\hat{p}, \hat{q}, \xi)$ that takes parameters $\hat{p}$, $\hat{q}$ that are estimates for p and q, and the separation parameter ξ as inputs, and randomly creates noisy labels for (x, y) pair drawn from Gen:

- Draw an example (x, y) from the oracle Gen.
- If $y \in E$, assign label $\ell = 1$ with probability a_1 and $\ell = 0$ with probability $a_0 = 1 - a_1$; Otherwise, assign label $\ell = 1$ with probability b_1 and $\ell = 0$ with probability $b_0 = 1 - b_1$, where

$$a_0 = 1/2 + \frac{\xi(\hat{p} + \hat{q} - 2)}{4(\hat{q} - \hat{p})} \quad \text{and} \quad b_0 = 1/2 + \frac{\xi(\hat{p} + \hat{q})}{4(\hat{q} - \hat{p})} \tag{1}$$

- Output the labeled example (x, ℓ).

It's easy to check that both vectors (a_0, a_1) and (b_0, b_1) form valid probabilities over $\{0, 1\}$ (see the appendix for a proof).

As mentioned, we need to ensure the class-conditional noise rates to be below $1/2$. As a first step, we work out the noise rates of $\mathbf{Lab}$ in terms of the true probabilities p and q, and show that the "estimated" noise rates based on $\hat{p}$ and $\hat{q}$ are below $(1/2 - \xi/4)$.

3. In the work of Blum and Mitchell (1998), the authors showed that any CN learnable class is also learnable when the class-conditional noise rates satisfy $\eta_0 + \eta_1 < 1$. Our construction here will imply a more general result—the class remains learnable when the noise rates satisfy $\eta_0 + \eta_1 \neq 1$.

Lemma 6 *Given a fixed ξ-distinguishing event E, the class-conditional noise rates of* $\mathbf{Lab}$ *are*

$$\eta_1 = \Pr[\ell = 0 \mid c(x) = 1] = qa_0 + (1-q)b_0 \qquad \textit{and} \qquad \eta_0 = \Pr[\ell = 1 \mid c(x) = 0] = pa_1 + (1-p)b_1.$$

Moreover, given any input estimates $(\hat{p}, \hat{q})$ for (p, q), the parameters a_0, a_1, b_0 and b_1 satisfy:

$$\hat{q}a_0 + (1-\hat{q})b_0 = \hat{p}a_1 + (1-\hat{p})b_1 \leq 1/2 - \xi/4.$$

By Lemma 6, we know that as long as the input estimates $\hat{p}$ and $\hat{q}$ are sufficiently close to p and q, the noise rates will be less than $1/2$. To obtain such estimates, we will guess the values of p and q on a grid of size $\lceil 1/\Delta \rceil^2$ in the range of $[0,1]^2$, where $\Delta \in [0,1]$ is some discretization parameter. Note that for some pair of values $i, j \in [\lceil 1/\Delta \rceil]$ and $i \neq j$ such that the guesses $(\hat{p}, \hat{q}) = (i\Delta, j\Delta)$ satisfies

$$\hat{p} \in [p - \Delta, p + \Delta] \qquad \text{and} \qquad \hat{q} \in [q - \Delta, q + \Delta]$$

Given such accurate guesses $\hat{p}$ and $\hat{q}$, we can then guarantee low noise rates as derived below:

Lemma 7 *Fix any $\Delta \in [0,1]$. Suppose that the estimates $\hat{p}$ and $\hat{q}$ satisfy $|p - \hat{p}| \leq \Delta$ and $|q - \hat{q}| \leq \Delta$, then the class-conditional noise rates η_0 and η_1 for $\mathbf{Lab}(\hat{p}, \hat{q}, \xi)$ are upper bounded by $1/2 - \xi/4 + \Delta$.*

Thus, if we choose the discretization parameter Δ to be below $\xi/4$, then the algorithm $\mathbf{Lab}(\hat{p}, \hat{q})$ is a valid example oracle $EX^{\eta}_{\mathrm{CCCN}}$ for some pair of guess estimates. Furthermore, if we apply the corresponding CCCN learning algorithm to the instantiations of $\mathbf{Lab}(\hat{p}, \hat{q})$ over all guesses $(\hat{p}, \hat{q})$, the output list of hypotheses is then guaranteed to contain an accurate one.

Lemma 8 *Let $\varepsilon, \delta \in (0,1)$. Suppose that the concept class $\mathcal{C}$ is CN learnable, and there exists an identified ξ-distinguishing event E for the two target distributions P_0 and P_1. Then there exists an algorithm $\mathcal{L}_1$ such that when given ε, δ, ξ and E as inputs, it will halt in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, 1/\xi, n)$, and with probability at least $1 - \delta$, output a list of hypotheses that contains some h such that $err(h) \leq \varepsilon$.*

In the next two sections, we will use the algorithm in Lemma 8 as a subroutine for learning the target concept c in the PwD framework.

4. Forward Reduction

Now we will give our forward algorithmic reduction: first use a CN learner to approximate the target concept c sufficiently well to separate the distributions P_0 and P_1, then learn each distribution using a distribution learner.[4] We will rely on the result in Section 3 to learn c with a CCCN learner, but we do not assume the learner has a priori identified a distinguishing event. Instead, we will assume that the distribution class $\mathcal{P}$ admits a parametric class of distinguishing events of polynomial size, which allows us to distinguish any two distributions in $\mathcal{P}$ with large KL-divergence.

4. We use the term "forward" to indicate that the reduction decomposes the learning process into the steps suggested by the generative model depicted in Figure 1.

Assumption 9 (Parametric Class of Distinguishing Events) *There exists a parametric class of events $\mathcal{E}(\cdot)$ for the distribution class $\mathcal{P}$ such that for any $\gamma > 0$ and for any two probability distributions P and Q in $\mathcal{P}$ with $\mathrm{KL}(P||Q) \geq \gamma$, the class of events $\mathcal{E}(\gamma)$ contains a ξ-distinguishing event E for P and Q, where $\xi \geq 1/\operatorname{poly}(k, 1/\gamma)$. Furthermore, $\mathcal{E}(\gamma)$ can be computed in time $\operatorname{poly}(k, 1/\gamma)$ and the cardinality $|\mathcal{E}(\gamma)| \leq \operatorname{poly}(k, 1/\gamma)$.*

To illustrate the intuition of how to construct such class of distinguishing events, we will give a simple example here. In the appendix, we will extend the construction to work for the class of spherical Gaussian distributions and product distributions over discrete domains.

Simple Example Consider the outcome space $\mathcal{Y} = \{0, 1\}^k$ and the class of full-support product distributions $\mathcal{P}$ over $\mathcal{Y}$. Let $P, Q \in \mathcal{P}$ be two distribution such that $\mathrm{KL}(P||Q) \geq \gamma$. Under the boundedness condition in Assumption 4, it can be shown that there exists some coordinate l such that $|P^l - Q^l| \geq 1/\operatorname{poly}(k, 1/\gamma)$, where $P^l = \Pr_{y\sim P}[y_l = 1]$ and $Q^l = \Pr_{y\sim Q}[y_l = 1]$. Therefore, for each coordinate l, the event that the coordinate y_j is 1 is a candidate distinguishing event, so the class of events is simply $\mathcal{E} = \{\mathbf{1}[y_l = 1] \mid l \in [k]\}$.

Here is our main result in the forward reduction.

Theorem 10 ((Formal version of Theorem 1)) *Under the Assumption 9 that $\mathcal{P}$ admits a parametric class of events $\mathcal{E}$, the joint class $(\mathcal{C}, \mathcal{P})$ is PwD-learnable as long as the concept class $\mathcal{C}$ is CN learnable, and the distribution class $\mathcal{P}$ is efficiently learnable.*

We will present our reduction in three key steps.

1. First, as a simple extension to Section 3, we can learn a hypothesis h with sufficiently small error assuming the class of events $\mathcal{E}$ contains a distinguishing event for the distributions P_0 and P_1.

2. Suppose we have learned an accurate hypothesis h from the first step, we can then use h to separate outcomes y drawn from P_0 and P_1, and apply the distribution learner to learn accurate distributions $\hat{P}_0$ and $\hat{P}_1$. This creates an accurate hypothesis model $\hat{T} = (h, \hat{P}_0, \hat{P}_1)$.

3. Finally, we need to handle the case where the distributions P_0 and P_1 are arbitrarily close, and there is no distinguishing event for us to learn the concept c. We will show in this case it is not necessary to learn the target concept, and we can directly learn the distributions without relying on an accurate hypothesis h.

The main technical challenge lies in the second and third steps, where we will apply the distribution learner (for single distributions in $\mathcal{P}$) on samples drawn from a mixture of P_0 and P_1. To tackle this issue, we will prove a robustness result for any distribution learner — as long as the input distribution is sufficiently close to the target distribution, the output distribution by the learner remains accurate. [5]

5. Our result actually extends to any PAC learning algorithm, and we omit the simple details.

4.1. CN Learning with a Class of Events

As a first step in our reduction, we will simply extend Lemma 8: for each event E in the event class $\mathcal{E}$, run the CCCN learner using E as a candidate distinguishing event. If the two target distributions P_0 and P_1 have large KL divergence, then one of the output hypotheses h will be accurate with respect to c:

Lemma 11 *Let $\varepsilon, \delta \in (0,1)$ and $\gamma > 0$. Suppose that the class $\mathcal{C}$ is CN learnable, the class $\mathcal{P}$ admits a parametric class of events $\mathcal{E}$ (as in Assumption 9). If the two distributions P_0 and P_1 satisfy $\max\{\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0)\} \geq \gamma$, then there exists an algorithm $\mathcal{L}_2$ that given sample access to Gen and $\varepsilon, \delta, \gamma$ as inputs, runs in time $\mathrm{poly}(1/\varepsilon, 1/\delta, 1/\gamma, n)$, and with probability at least $1-\delta$ outputs a list of hypotheses H that contains a hypothesis h with error $err(h) \leq \varepsilon$.*

4.2. Robustness of Distribution Learner

Before we proceed to the next two steps of the reduction, we will briefly digress to give a useful robustness result showing that the class $\mathcal{P}$ remains efficiently learnable even if the input distribution is slightly perturbed. Our result relies on the well-known *Le Cam's method*, which is a powerful tool for giving lower bounds in hypothesis testing. We state the following version for our purpose.[6]

Lemma 12 *[Le Cam's method (see e.g. Le Cam (1986); Yu (1997))] Let Q_0 and Q_1 be two probability distributions over $\mathcal{Y}$, and let $\mathcal{A}\colon \mathcal{Y}^m \to \{0,1\}$ be a mapping from m observations in $\mathcal{Y}$ to either 0 or 1. Then*

$$\Pr_{\mathcal{A}, Y^m \sim Q_0^m}[\mathcal{A}(Y^m) \neq 0] + \Pr_{\mathcal{A}, Y^m \sim Q_1^m}[\mathcal{A}(Y^m) \neq 1] \geq 1 - \sqrt{m\mathrm{KL}(Q_0||Q_1)/2}$$

where $Y^m \sim Q_\theta^m$ denotes an i.i.d. sample of size m drawn from the distribution Q_θ.

The lemma above shows that any statistical procedure that determines whether the underlying distribution is Q_0 or Q_1 based on m independent observations must have high error if the two distributions are too close. In particular, if their KL divergence satisfies $\mathrm{KL}(Q_0||Q_1) \leq 1/m$, then the procedure has at least constant error probability under measure Q_0 or Q_1. Now let's construct such a procedure $\mathcal{A}$ using any distribution learner $\mathcal{L}$ for the class $\mathcal{P}$. Suppose the learner is ε-accurate with high probability when given sample of size m, and the distribution Q_0 is in the class $\mathcal{P}$. Consider the following procedure $\mathcal{A}$:

- Run the learning algorithm $\mathcal{L}$ on sample S of size m. If the algorithm fails to output a hypothesis distribution, output 1. Otherwise, let $\hat{Q}$ be the output distribution by $\mathcal{L}$.
- If $\mathrm{KL}(Q_0||\hat{Q}) \leq \varepsilon$, output 0; otherwise output 1.

Note that if the sample S is drawn from the distribution Q_0, then $\mathcal{A}$ will correctly output 0 with high probability based on the accuracy guarantee of $\mathcal{L}$. This means the procedure has to err when S is drawn from the slightly perturbed distribution Q_1, and so the learner will with constant probability output an accurate distribution $\hat{Q}$ such that $\mathrm{KL}(Q_0||\hat{Q}) \leq \varepsilon$. More formally:

6. In the usual statement of Le Cam's method, the right-hand side of the inequality is in fact $1 - \|Q_0^m - Q_1^m\|_{tv}$, where $\|\cdot\|_{tv}$ denotes total variation distance. We obtain the current bound by a simple application of Pinsker inequality.

Lemma 13 *Let $\varepsilon > 0$, $\delta \in (0, 1/2)$ and $m \in \mathbb{N}$. Suppose there exists a distribution learner $\mathcal{L}$ such that for any unknown target distribution $P \in \mathcal{P}$, when $\mathcal{L}$ inputs m random draws from P, it with probability at least $1 - \delta$ outputs a distribution $\hat{P}$ such that $\mathrm{KL}(P||\hat{P}) \leq \varepsilon$. Then for any $Q_0 \in \mathcal{P}$ and any distribution Q_1 over the same range $\mathcal{Y}$, if the learner $\mathcal{L}$ inputs a sample of size m drawn independently from Q_1, it will with probability at least $1 - \delta'$ output a distribution $\hat{Q}$ such that $\mathrm{KL}(Q_0||\hat{Q}) \leq \varepsilon$, where $\delta' = \delta + \sqrt{m\mathrm{KL}(Q_0||Q_1)/2}$.*

Proof Consider the procedure $\mathcal{A}$ constructed above that uses the learner $\mathcal{L}$ as a subroutine. By the guarantee of the algorithm, we know that $\Pr_{\mathcal{L}, Y^m \sim Q_0^m}[\mathrm{KL}(Q_0||\hat{Q}) \leq \varepsilon] \geq 1 - \delta$. This means

$$\Pr_{\mathcal{A}, Y^m \sim Q_0^m}[\mathcal{A}(Y^m) \neq Q_0] \leq \delta.$$

By Lemma 12, we have

$$\Pr_{\mathcal{A}, Y^m \sim Q_1^m}[\mathcal{A}(Y^m) \neq Q_1] \geq 1 - \sqrt{\frac{m}{2}\mathrm{KL}(Q_0||Q_1)} - \delta.$$

This in turn implies that with probability at least $(1 - \delta - \sqrt{\frac{m}{2}\mathrm{KL}(Q_0||Q_1)})$ over the draws of $Y^m \sim Q_1^m$ and the internal randomness of $\mathcal{L}$, the output distribution $\hat{Q}$ satisfies $\mathrm{KL}(P||\hat{Q}) \leq \varepsilon$. ■

Therefore, if the KL divergence between the target distribution and the input distribution is smaller than inverse of the (polynomial) sample size, the output distribution by the learner is accurate with constant probability. By using a standard amplification technique, we can guarantee the accuracy with high probability:

Lemma 14 *Suppose that the distribution class $\mathcal{P}$ is PAC learnable. There exist an algorithm $\mathcal{L}_2$ and a polynomial $m_{\mathcal{P}}(\cdot, \cdot, \cdot)$ such that that for any target unknown distribution P, when given any $\varepsilon > 0$ and $0 < \delta \leq 1/4$ as inputs and sample access from a distribution Q such that $\mathrm{KL}(P||Q) \leq 1/(2m_{\mathcal{P}}(1/\varepsilon, 1/\delta, k))$, runs in time $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$ and outputs a list of distributions $\mathcal{P}'$ that with probability at least $1 - \delta$ contains some $\hat{P} \in \mathcal{P}'$ with $\mathrm{KL}(P||\hat{P}) \leq \varepsilon$.*

As a consequence, even when input sample distribution is slightly "polluted", we can still learn the target distribution accurately with a small blow-up in the computational and sample complexity.

4.3. Learning the Distributions with an Accurate Hypothesis

Now we will return to the second step of our reduction: use an accurate hypothesis h and distribution learner for $\mathcal{P}$ to learn the two distributions P_0 and P_1. For any observation (x, y) drawn from the example oracle Gen, we can use the hypothesis h to determine whether the outcome y is drawn from P_0 or P_1, which allows us to create independent samples from both distributions. However, because of the small error of h with respect to the target concept c, the input sample is in fact drawn from a mixture between P_0 and P_1. To remedy this problem, we will choose a sufficiently small error rate for hypothesis h (but still an inverse polynomial in the learning parameters), which guarantees that the mixture is close enough to either one of single target distributions. We can then apply the result in Lemma 14 to learn each distribution, which together gives us a hypothesis model $(h, \hat{P}_0, \hat{P}_1)$.

Lemma 15 *Suppose that the distribution class $\mathcal{P}$ is efficiently learnable. Let $\varepsilon > 0, 0 < \delta \leq 1$ and $h \in \mathcal{C}$ be an hypothesis. Then there exists an algorithm $\mathcal{L}_3$ and a polynomial $r(\cdot,\cdot,\cdot)$ such that when given ε, δ and h as inputs, $\mathcal{L}_3$ runs in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$, and outputs a list of probability models $\mathcal{T}$ such that with probability at least $1-\delta$ there exists some $\hat{T} \in \mathcal{T}$ such that $err(\hat{T}) \leq \varepsilon$, as long as the hypothesis h satisfies $err(h) \leq 1/r(1/\varepsilon, 1/\delta, k)$.*

4.4. Directly Applying the Distribution Learner

In the last step of our forward reduction, we will consider the case where the two target distributions P_0 and P_1 are too close to admit a distinguishing event, and so we will not be able to learn the target concept c as in the first step. We show that in this case learning c is not necessary for obtaining an accurate probability model — we can simply run the robust distribution learner developed in Lemma 14 over the samples drawn from the mixture to learn single distribution.

We will first define the following notion of *healthy mixture*, which captures the mixture distributions with non-trivial weights on two sufficiently different components. This will also facilitate our discussion in the reverse reduction.

Definition 16 (Healthy Mixture) *Let Q be mixture of two distributions Q_0 and Q_1 from the class $\mathcal{P}$, and let w_0 and w_1 be the weights on the two components respectively. Then Q is an* η-healthy mixture *if both $\min\{w_0, w_1\} \geq \eta$ and $\max\{\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0)\} \geq \eta$ hold. If one of the two conditions does not hold, we will call Q an* η-unhealthy mixture.

We now show that whenever the mixture distribution P is unhealthy, we can use the robust learner in Lemma 14 to directly learn a distribution $\hat{P}$ for our prediction purpose (simply always predict with $\hat{P}$ regardless of the context x). Note that this not only includes the case where P_0 and P_1 are arbitrarily close, but also the one where the weight on one component is close to 0, which will be useful in Section 5.

Lemma 17 *Suppose that the distribution class $\mathcal{P}$ is PAC learnable. Let P be the unconditional mixture distribution over the outcomes $\mathcal{Y}$ under the distribution Gen. Let $\varepsilon > 0$ and $\delta \in (0,1)$. Then there exists an algorithm $\mathcal{L}_4$ and a polynomial $g(\cdot,\cdot,\cdot)$ such that when $\mathcal{L}_4$ is given sample access to Gen and ε, δ as inputs, it runs in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$ and it will with probability at least $1-\delta$, output a list of distributions $\mathcal{P}'$ that contains $\hat{P}$ with $\mathbb{E}_{x\sim\mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P})\right] \leq \varepsilon$, as long as P is an η-unhealthy mixture for some $\eta \leq 1/g(k, 1/\varepsilon, 1/\delta)$.*

We will now combine the all the tools to provide a proof sketch for Theorem 10 (see the appendix for details).

Proof [Proof Sketch for Theorem 10] Our algorithm for PwD learning the joint class $(\mathcal{C}, \mathcal{P})$ is roughly the following. First, we will make use of Assumption 9 and obtain a set of candidate distinguishing events for the target distributions P_0 and P_1. We will run the CCCN learner to learn c using each candidate event E to generate noisy labels. This generates a list of hypotheses. We will use the hypotheses h to separate the two distributions P_0 and P_1 and apply the algorithm in Lemma 15 to learn each distribution individually. This will give polynomially many hypothesis models $\hat{T} = (h, \hat{P}_0, \hat{P}_1)$. By Lemma 11 and Lemma 15, we know at least one of the models is accurate when P_0 and P_1 are sufficiently different.

To cover the case where the two distributions are too close, we will use the algorithm in Lemma 17 to learn a list of distributions over $\mathcal{Y}$. In particular, the model $(h', \hat{P}, \hat{P})$ is accurate for at least one of the output distribution $\hat{P}$.

Together, the two procedures above will give a list of polynomially many hypothesis models, at least one of which is guaranteed to be accurate. We will use the standard *maximum likehood method* to output the model that minimizes empirical log-loss, and with high probability, this will be an accurate model.[7] ■

We previously gave examples (such as product distributions and special cases of multivariate Gaussians) that admit small classes of distinguishing events, and to which Theorem 10 can be applied. There are other important cases — such as general multivariate Gaussians — for which we do not know such classes.[8] However, we now describe a different, "reverse" reduction that instead assumes learnability of mixtures, and thus is applicable to more general Gaussians via known mixture learning algorithms (Dasgupta, 1999; Arora and Kannan, 2001; Feldman et al., 2006).

5. Reverse Reduction

In our reverse reduction, our strategy is to first learn the two distributions P_0 and P_1 sufficiently well, and then construct a specialized distinguishing event to learn the target concept c with a CCCN learner.[9] We will make a stronger learnability assumption on the distribution class $\mathcal{P}$ — we assume a *parametrically correct* learner for any healthy mixture of two distributions in $\mathcal{P}$.

Assumption 18 (Parametrically Correct Mixture Learning) *There exists a mixture learner $\mathcal{L}_M$ and a polynomial ρ such that for any $\varepsilon > 0, 0 < \delta \leq 1$, and for any Z that is an η-healthy mixture of two distributions Y_0 and Y_1 from $\mathcal{P}$, the following holds: if $\mathcal{L}_M$ is given sample access to Z and $\varepsilon, \delta > 0$ as inputs, $\mathcal{L}_M$ runs in time $\mathrm{poly}(k, 1/\varepsilon, 1/\delta)$ and with probability at least $1 - \delta$, outputs a mixture $\hat{Z}$ of distributions $\hat{Y}_0$ and $\hat{Y}_1$ such that $\max\{\mathrm{KL}(Y_0||\hat{Y}_0), \mathrm{KL}(Y_1||\hat{Y}_1)\} \leq \varepsilon$.*

We remark that the assumption of *parametric correctness* is a mild condition, and is satisfied by almost all mixture learning algorithms in the literature (see e.g. Dasgupta (1999); Feldman et al. (2006, 2008); Hsu and Kakade (2013)). Also note that we only require this condition when the healthy mixture condition in Theorem 16 is met. If the two either the two distributions Y_0 and Y_1 are arbitrarily close or the mixture is extremely unbalanced, we are not supposed to learn both components correctly.

Theorem 19 (Formal Version of Theorem 2) *Suppose the class $\mathcal{C}$ is CN learnable, the distribution class $\mathcal{P}$ is efficiently learnable and satisfies the parametrically correct mixture learning assumption (Assumption 18). Then the joint class $(\mathcal{C}, \mathcal{P})$ is PwD-learnable.*

With the tools we develop for the forward reduction, the proof for reverse reduction is straightforward. There are essentially two cases we need to deal with. In the first case where the mixture distribution over $\mathcal{Y}$ is healthy, we can use the parametrically correct mixture learner to learn the two

7. See the appendix for the details and analysis of the maximum likelihood method in the PwD model.
8. We conjecture that Gaussians do indeed have a small set of distinguishing events, but have not been able to prove it.
9. We use the term "reverse" to indicate that the reduction decomposes the learning process into the steps suggested by the inverted generative model depicted in Figure 2.

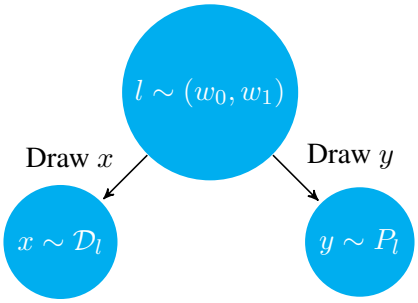

Figure 2: An alternative view of the generative model Gen: first draw a Bernoulli label l with bias $w_1 = \Pr_{\mathcal{D}}[c(x) = 1]$, then draw a context x from the conditional distribution $\mathcal{D}_l$ on $c(x) = l$, and an outcome y from the distribution P_l. In the forward reduction, we first learn the concept c over $\mathcal{X}$ (which determines the label l), so we can separate the data and learn each distribution using a (single) distribution learner. In the reverse reduction, we will first use the mixture learner to learn both P_0 and P_1, and then use such information to obtain estimates for the label l for learning the concept c.

target distributions, we can then use the accurate approximations $\hat{P}_0$ and $\hat{P}_1$ to find a distinguishing event for P_0 and P_1, which allows us to learn the concept c with a CCCN learner. In the case where the mixture distribution is unhealthy and we cannot learn the components accurately, we can again appeal to the robustness result we show using Le Cam's method — we can directly apply the learner for single distributions and learn P_0 or P_1.

5.1. CN Learning with a Mixture Learner

Given any two distributions P, Q over $\mathcal{Y}$ and a parameter τ, consider the event (or subset)

$$E(P, Q, \tau) = \{y \in \mathcal{Y} \mid P(y) \geq 2^{\tau}\, Q(y)\}$$

We will first show that such subset is a distinguishing event for the input distributions P and Q as long as the distributions P and Q are sufficiently different.

Lemma 20 *Fix any $\gamma \in (0, 1]$. Suppose that $\mathrm{KL}(P||Q) \geq \gamma$, then $E(P, Q, \gamma/2)$ is a $(\gamma^2/(8M))$-distinguishing event for the distributions P and Q.*

Next, we show that even if we only have access to the approximate distributions $\hat{P}$ and $\hat{Q}$, we can still identify a distinguishing event for P and Q, as long as the approximations are accurate.

Lemma 21 *Suppose that the distributions $P, \hat{P}, Q, \hat{Q}$ over $\mathcal{Y}$ satisfy that $\mathrm{KL}(P||\hat{P}) \leq \alpha$, $\mathrm{KL}(Q||\hat{Q}) \leq \alpha$, and $\mathrm{KL}(P||Q) \geq \gamma$ for some $\alpha, \gamma \in (0, 1]$. Then the event $E(\hat{P}, \hat{Q}, (\gamma^2/(8M) - \sqrt{2\alpha})^2)$ is a ξ-distinguishing event with $\xi \geq 1/\operatorname{poly}(1/\gamma, 1/\alpha, k)$ as long as $\gamma > 8M(\sqrt{2\alpha} + (8M^2\alpha)^{1/8})$.*

Given these structural lemmas, we now know a way to construct a distinguishing event based on approximations to the target distributions P_0 and P_1. We can then create a and use the algorithm in Lemma 8 to learn the concept c, and in turn compute a list of hypothesis models, one of which is guaranteed to be accurate when the mixture distribution is healthy.

Lemma 22 *Suppose the class $\mathcal{P}$ satisfies the parametric mixture learning assumption (Assumption 18), the class $\mathcal{C}$ is CN learnable, and mixture distribution over $\mathcal{Y}$ is γ-healthy for some $\gamma > 0$. Then there exists an algorithm $\mathcal{L}$ that given ε, δ and γ as inputs and sample access from Gen, halts in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, 1/\gamma, n, k)$, and with probability at least $1 - \delta$, outputs a list of probability models $\mathcal{T}$ that contains some $\hat{T}$ with $err(\hat{T}) \leq \varepsilon$.*

Finally, to wrap up and prove Theorem 19, we also need to handle the case where healthy mixture condition in Theorem 16 does not hold. We will again appeal to the robust distribution learner in Lemma 17 to learn the distributions directly, and construct hypothesis models based on the output distributions. To guarantee that the output hypothesis model is accurate, we will again use the maximum likelihood method to select the model with the minimum empirical log-loss (formal proof deferred to the appendix).

6. Future Work

Despite the generality of our results and reductions, there remain some appealing directions for further research. These include allowing the conditioning event to be richer than a simple binary function $c(x)$, for instance multi- or even real-valued. This might first entail the development of theories for noisy learning in such models, which is well-understood primarily in the binary setting.

We also note that our study has suggested an interesting problem in pure probability theory, namely whether general Gaussians permit a small class of distinguishing events.

Acknowledgments

We thank We thank Akshay Krishnamurthy and Shahin Jabbari for helpful discussions.

References

Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2 (4):343–370, 1987. doi: 10.1007/BF00116829. URL http://dx.doi.org/10.1007/BF00116829.

Sanjeev Arora and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC '01, pages 247–257, New York, NY, USA, 2001. ACM. ISBN 1-58113-349-9. doi: 10.1145/380752.380808. URL http://doi.acm.org/10.1145/380752.380808.

Avrim Blum and Tom M. Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998.*, pages 92–100, 1998. doi: 10.1145/279943.279962. URL http://doi.acm.org/10.1145/279943.279962.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *40th Annual Symposium on Foundations of Computer Science, FOCS '99, 17-18 October, 1999, New York, NY, USA*, pages 634–644, 1999. doi: 10.1109/SFFCS.1999.814639. URL http://dx.doi.org/10.1109/SFFCS.1999.814639.

Scott E. Decatur. PAC learning with constant-partition classification noise and applications to decision tree induction. In *Proceedings of the Fourteenth International Conference on Machine Learning*, ICML '97, pages 83–91, San Francisco, CA, USA, 1997. Morgan Kaufmann Publishers Inc. ISBN 1-55860-486-3. URL http://dl.acm.org/citation.cfm?id=645526.657273.

Jon Feldman, Rocco A. Servedio, and Ryan O'Donnell. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *Learning Theory, 19th Annual Conference on Learning Theory, COLT 2006, Pittsburgh, PA, USA, June 22-25, 2006, Proceedings*, pages 20–34, 2006. doi: 10.1007/11776420_5. URL http://dx.doi.org/10.1007/11776420_5.

Jon Feldman, Ryan O'Donnell, and Rocco A. Servedio. Learning mixtures of product distributions over discrete domains. *SIAM J. Comput.*, 37(5):1536–1564, 2008. doi: 10.1137/060670705. URL http://dx.doi.org/10.1137/060670705.

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 http://dx.doi.org/10.1016/0890-5401(92)90010-D.

Daniel J. Hsu and Sham M. Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Innovations in Theoretical Computer Science, ITCS '13, Berkeley, CA, USA, January 9-12, 2013*, pages 11–20, 2013. doi: 10.1145/2422436.2422439. URL http://doi.acm.org/10.1145/2422436.2422439.

Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, January 1994. ISSN 0004-5411. doi: 10.1145/174644.174647. URL http://doi.acm.org/10.1145/174644.174647.

Michael Kearns, Ming Li, and Leslie Valiant. Learning boolean formulas. *J. ACM*, 41(6):1298–1328, November 1994a. ISSN 0004-5411. doi: 10.1145/195613.195656. URL http://doi.acm.org/10.1145/195613.195656.

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

Michael J. Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the Twenty-Sixth Annual ACM Symposium on Theory of Computing, 23-25 May 1994, Montréal, Québec, Canada*, pages 273–282, 1994b. doi: 10.1145/195058.195155. URL http://doi.acm.org/10.1145/195058.195155.

Lucien Marie Le Cam. *Asymptotic methods in statistical decision theory*. Springer series in statistics. Springer-Verlag, New York, 1986. ISBN 0-387-96307-3. URL http://opac.inria.fr/record=b1093333.

Leonard Pitt and Manfred K. Warmuth. Prediction-preserving reducibility. *J. Comput. Syst. Sci.*, 41(3):430–467, December 1990. ISSN 0022-0000. doi: 10.1016/0022-0000(90)90028-J. URL http://dx.doi.org/10.1016/0022-0000(90)90028-J.

Liva Ralaivola, François Denis, and Christophe Nicolas Magnan. CN = CPCN. In *Machine Learning, Proceedings of the Twenty-Third International Conference (ICML 2006), Pittsburgh, Pennsylvania, USA, June 25-29, 2006*, pages 721–728, 2006. doi: 10.1145/1143844.1143935. URL http://doi.acm.org/10.1145/1143844.1143935.

Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5(2):197–227, July 1990. ISSN 0885-6125. doi: 10.1023/A:1022648800760. URL http://dx.doi.org/10.1023/A:1022648800760.

Leslie G. Valiant. A theory of the learnable. In *Proceedings of the 16th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1984, Washington, DC, USA*, pages 436–445, 1984. doi: 10.1145/800057.808710. URL http://doi.acm.org/10.1145/800057.808710.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, 68(4):841–860, 2004. doi: 10.1016/j.jcss.2003.11.008. URL http://dx.doi.org/10.1016/j.jcss.2003.11.008.

Bin Yu. Assouad, fano, and le cam. In *Festschrift for Lucien Le Cam*, pages 423–435. Springer New York, 1997.

Appendix A. Missing Details and Proofs

A.1. Missing Details in Section 2

Definition 23 (CN Learnability (Angluin and Laird, 1987)) *Let $\mathcal{C}$ be a concept class over $\mathcal{X}$. We say that $\mathcal{C}$ is efficiently learnable with noise (CN learnable) if there exists a learning algorithm $\mathcal{L}$ such that for any $c \in \mathcal{C}$, any distribution $\mathcal{D}$ over $\mathcal{X}$, any noise rate $0 \leq \eta < 1/2$, and for any $0 < \varepsilon \leq 1$ and $0 < \delta \leq 1$, the following holds: if $\mathcal{L}$ is given inputs η_b (where $1/2 > \eta_b \geq \eta$), ε, δ, n, and is given access to $EX^{\eta}_{\mathrm{CN}}(c, \mathcal{D})$, then $\mathcal{L}$ will halt in time bounded by $\mathrm{poly}(1/(1-2\eta_b), 1/\varepsilon, 1/\delta, n)$ and output a hypothesis $h \in \mathcal{C}$ that with probability at least $1-\delta$ satisfies $err(h) \leq \varepsilon$.*

Lemma 24 (CN = CCCN (Ralaivola et al., 2006)) *Suppose that the concept class $\mathcal{C}$ is CN learnable. Then there exists an algorithm $\mathcal{L}_{\mathcal{C}}$ and a polynomial $m_{\mathcal{C}}(\cdot,\cdot,\cdot,\cdot)$ such that for every target concept $c \in \mathcal{C}$, any $\varepsilon, \delta \in (0,1]$, for any noise rates $\eta_0, \eta_1 \leq \eta_b < 1/2$, if L is given inputs $\varepsilon, \delta, \eta_b$ and access to $EX^{\eta}_{\mathrm{CCCN}}(c, \mathcal{D})$, then L will halt in time bounded by $m_{\mathcal{C}}(1/(1-2\eta_b), 1/\varepsilon, 1/\delta, n)$, and output with probability at least $1-\delta$ a hypothesis h with error $err(h) \leq \varepsilon$. We will say that $\mathcal{L}_{\mathcal{C}}$ is an (efficient) CCCN learner for $\mathcal{C}$ with sample complexity $m_{\mathcal{C}}$.*

Definition 25 (Evaluator (Kearns et al., 1994b)) *Let $\mathcal{P}$ be a class of distributions over the outcome space $\mathcal{Y}$. We say that $\mathcal{P}$ has a efficient* evaluator *if there exists a polynomial p such that for any $n \geq 1$, and for any distribution $P \in \mathcal{P}$, there exists an algorithm E_P with runtime bounded by $\mathrm{poly}(k)$ that given an input $y \in \mathcal{Y}$ outputs the probability (density) assigned to y by P. Thus, if $y \in \mathcal{Y}$, then $E_P(y)$ is the weight of y under P. We call E_P an* evaluator *for P.*

A.2. Missing Proofs in Section 3

Claim 26 *The values of a_0 and b_0 satisfy $a_0, b_0 \in [0,1]$.*

Proof Without loss of generality, let's assume that $q \geq p + \xi$. Since $p + q \in [0,2]$, we know that $a_0 \leq 1/2$ and we can write

$$a_0 = 1/2 + \frac{\xi(p+q-2)}{4(q-p)} \geq 1/2 - \frac{\xi}{2(q-p)} \geq 1/2 - 1/2 \geq 0$$

Similarly, we know that $b_0 \geq 1/2$ and we can write

$$b_0 = 1/2 + \frac{\xi(p+q)}{4(q-p)} \leq 1/2 + \frac{\xi/2}{\xi} = 1$$

This proves our claim. ■

Lemma 27 *Given a fixed ξ-distinguishing event E, the class-conditional noise rates of* **Lab** *are*

$$\eta_1 = \Pr[\ell = 0 \mid c(x) = 1] = qa_0 + (1-q)b_0 \qquad \textit{and} \qquad \eta_0 = \Pr[\ell = 1 \mid c(x) = 0] = pa_1 + (1-p)b_1.$$

Moreover, given any input estimates $(\hat{p}, \hat{q})$ for (p, q), the parameters a_0, a_1, b_0 and b_1 satisfy:

$$\hat{q}a_0 + (1-\hat{q})b_0 = \hat{p}a_1 + (1-\hat{p})b_1 \leq 1/2 - \xi/4.$$

Proof We can derive the probabilities as follows

$$\begin{aligned}\Pr[\ell = 0 \mid c(x) = 1] &= \Pr[(\ell = 0) \wedge (y \in E) \mid c(x) = 1] + \Pr[(\ell = 0) \wedge (y \notin E) \mid c(x) = 1] \\ &= \Pr_{\text{Gen}}[y \in E \mid c(x) = 1] \Pr_{\mathbf{Lab}}[\ell = 0 \mid (y \in E) \wedge (c(x) = 1)] \\ &+ \Pr_{\text{Gen}}[y \notin E \mid c(x) = 1] \Pr_{\mathbf{Lab}}[\ell = 0 \mid (y \notin E) \wedge (c(x) = 1)] \\ &= \Pr_{\text{Gen}}[y \in E \mid c(x) = 1] a_0 + \Pr_{\text{Gen}}[y \notin E \mid c(x) = 1] b_0 \\ &= q\, a_0 + (1-q) b_0\end{aligned}$$

Similarly, we can also show that $\Pr[\ell = 1 \mid c(x) = 0] = pa_1 + (1-p)b_1$. For the second part of the statement, we can show

$$\hat{q}a_0 + (1-\hat{q})b_0 = \frac{\hat{q}}{2} + \frac{\xi(\hat{p}+\hat{q}-2)\hat{q}}{4(\hat{q}-\hat{p})} + \frac{(1-\hat{q})}{2} + \frac{\xi(\hat{p}+\hat{q})(1-\hat{q})}{4(\hat{q}-\hat{p})} = 1/2 - \xi/4$$
$$\hat{p}a_1 + (1-\hat{p})b_1 = \frac{\hat{p}}{2} - \frac{\xi(\hat{p}+\hat{q}-2)\hat{p}}{4(\hat{q}-\hat{p})} + \frac{(1-\hat{p})}{2} - \frac{\xi(\hat{p}+\hat{q})(1-\hat{p})}{4(\hat{q}-\hat{p})} = 1/2 - \xi/4$$

which recovers our claim. ■

Lemma 28 *Fix any $\Delta \in [0,1]$. Suppose that the estimates $\hat{p}$ and $\hat{q}$ satisfy $|p - \hat{p}| \le \Delta$ and $|q - \hat{q}| \le \Delta$, then the class-conditional noise rates η_0 and η_1 for $\mathbf{Lab}(\hat{p}, \hat{q}, \xi)$ are upper bounded by $1/2 - \xi/4 + \Delta$.*

Proof Since $a_0, a_1, b_0, b_1 \in [0,1]$, and by our assumption on the accuracy of $\hat{p}$ and $\hat{q}$, we have

$$\eta_1 - (\hat{q}a_0 + (1-\hat{q})b_0) = (qa_0 + (1-q)b_0) - (\hat{q}a_0 + (1-\hat{q})b_0) = (q-\hat{q})(a_0 - b_0) \le \Delta$$
$$\eta_0 - (\hat{q}a_1 + (1-\hat{q})b_1) = (qa_1 + (1-q)b_1) - (\hat{q}a_1 + (1-\hat{q})b_1) = (q-\hat{q})(a_1 - b_1) \le \Delta$$

The result of Lemma 6 tells us that

$$\hat{q}a_0 + (1-\hat{q})b_0 = \hat{p}a_1 + (1-\hat{p})b_1 \le 1/2 - \xi/4$$

Therefore, we must also have $\eta_0, \eta_1 \le 1/2 - \xi/4 + \Delta$. ■

Lemma 29 *Let $\varepsilon, \delta \in (0,1)$. Suppose that the concept class $\mathcal{C}$ is CN learnable, and there exists an identified ξ-distinguishing event E for the two target distributions P_0 and P_1. Then there exists an algorithm $\mathcal{L}_1$ such that when given ε, δ, ξ and E as inputs, it will halt in time bounded by $\text{poly}(1/\varepsilon, 1/\delta, 1/\xi, n)$, and with probability at least $1-\delta$, output a list of hypotheses that contains some h such that $err(h) \le \varepsilon$.*

Proof Since the concept class $\mathcal{C}$ is CN learnable, by the result of Ralaivola et al. (2006) we know there exists an efficient algorithm $\mathcal{A}$ that when given access to some example oracle EX^{η}_{CCCN} with $\eta_0, \eta_1 \le 1/2 - \xi/8$, outputs a hypothesis h with error bounded ε with probability at least $1-\delta$, halts in time $\text{poly}(1/\varepsilon, 1/\delta, 1/\xi, n)$.

Now let parameter $\Delta = \xi/8$, and consider the algorithm: for each pair of values $(\hat{p}, \hat{q}) = (i\Delta, j\Delta)$ such that $i, j \in [\lceil 1/\Delta \rceil]$ and $i \neq j$, use the $\mathbf{Lab}(\hat{p}, \hat{q}, \xi)$ to generate labeled examples, and run the algorithm $\mathcal{A}$ with sample access to $\mathbf{Lab}$; if the algorithm halts in time p and outputs an hypothesis $\hat{h}$, store the hypothesis in a the list H. In the end, output the hypothesis list.

By Lemma 7, we know for some guessed values of p' and q', the algorithm $\mathbf{Lab}(p', q', \xi)$ is an CCCN oracle with noise rates $\eta_0, \eta_1 \leq 1/2 - \xi/8$. Then by the guarantee of the learning algorithm, we know with probability at least $1 - \delta$, the algorithm will output an ε-accurate hypothesis under these guesses. ■

A.3. Missing Proofs in Section 4

Lemma 30 *Let $\varepsilon, \delta \in (0, 1)$ and $\gamma > 0$. Suppose that the class $\mathcal{C}$ is CN learnable, the class $\mathcal{P}$ admits a parametric class of events $\mathcal{E}$ (as in Assumption 9). If the two distributions P_0 and P_1 satisfy $\max\{\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0)\} \geq \gamma$, then there exists an algorithm $\mathcal{L}_2$ that given sample access to Gen and $\varepsilon, \delta, \gamma$ as inputs, runs in time $\mathrm{poly}(1/\varepsilon, 1/\delta, 1/\gamma, n)$, and with probability at least $1 - \delta$ outputs a list of hypotheses H that contains a hypothesis h with error $err(h) \leq \varepsilon$.*

Proof Consider the following algorithm. We will first use the oracle $\mathcal{E}$ with input parameter γ to obtain a class of events $\mathcal{E}(\gamma)$ that contains a ξ-distinguishing event E^* with $\xi \geq \mathrm{poly}(\gamma, 1/n)$. Then for each event $E \in \mathcal{E}(\gamma)$, we will run the algorithm $\mathcal{A}$ in Lemma 8 with accuracy parameters ε, δ, separation parameter ξ, and E as an hypothetical distinguishing event as input. For each event, the instantiation of algorithm $\mathcal{A}$ will halt in polynomial time. Furthermore, when the input event is E^* it will with probability at least $1 - \delta$ outputs a list of hypotheses H that contains a hypothesis h such that $err(h) \leq \varepsilon$ by the guarantee of Lemma 8. ■

Lemma 31 *Suppose that the distribution class $\mathcal{P}$ is PAC learnable. There exist an algorithm $\mathcal{L}_2$ and a polynomial $m_{\mathcal{P}}(\cdot, \cdot, \cdot)$ such that that for any target unknown distribution P, when given any $\varepsilon > 0$ and $0 < \delta \leq 1/4$ as inputs and sample access from a distribution Q such that $\mathrm{KL}(P||Q) \leq 1/(2m_{\mathcal{P}}(1/\varepsilon, 1/\delta, k))$, runs in time $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$ and outputs a list of distributions $\mathcal{P}'$ that with probability at least $1 - \delta$ contains some $\hat{P} \in \mathcal{P}'$ with $\mathrm{KL}(P||\hat{P}) \leq \varepsilon$.*

Proof Let $\mathcal{L}$ be a distribution learner that given a independent sample of size m drawn from the unknown target distribution P, runs in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, n)$ with probability at least $1 - \delta$, outputs a distribution P' such that $\mathrm{KL}(P||P') \leq \varepsilon$. By Lemma 13, we know that with probability at least $(1/2 - \delta) \geq 1/4$, the algorithm can also output a distribution P'' such that $\mathrm{KL}(P||P'') \leq \varepsilon$ if the algorithm is given a sample of size m drawn from the distribution Q.

Let $r = \log_{3/4}(1/\delta)$. Now we will run the algorithm r times on r independent samples, each of size m. Let $\mathcal{P}'$ be the list of output hypothesis distributions in these runs. We know that with probability at least $1 - (1 - 1/4)^r = 1 - \delta$, there exists a distribution $\hat{P} \in \mathcal{P}'$ such that $\mathrm{KL}(P||\hat{P}) \leq \varepsilon$. ■

The following is a technical lemma that allows us to bound the KL divergence between between a mixture distribution and one of its component.

Lemma 32 *Let P and Q be two distributions over $\mathcal{Y}$ and R be a mixture of P and Q with weights w_p and w_q respectively. Then we have* $\mathrm{KL}(P||R) \leq w_q \mathrm{KL}(P||Q)$.

Proof Let w_p and w_q be the weights associated with P and Q respectively in the mixture R.

$$\begin{aligned}
\mathrm{KL}(P||R) &= \int_y P(y) \log\left(\frac{P(y)}{R(y)}\right) dy \\
&= \int_y (w_p P(y) + w_q P(y)) \log\left(\frac{w_p P(y) + w_q P(y)}{w_p P(y) + w_q Q(y)}\right) dy \\
\text{(by the log-sum inequality)} \quad &\leq \int_y \left(w_p P(y) \log\left(\frac{w_p P(y)}{w_p P(y)}\right)\right) dy + \int_y \left(w_q P(y) \log\left(\frac{w_q P(y)}{w_q Q(y)}\right)\right) dy \\
&= w_q \mathrm{KL}(P||Q)
\end{aligned}$$

which proves our claim. ■

Lemma 33 *Suppose that the distribution class $\mathcal{P}$ is efficiently learnable. Let $\varepsilon > 0, 0 < \delta \leq 1$ and $h \in \mathcal{C}$ be an hypothesis. Then there exists an algorithm $\mathcal{L}_3$ and a polynomial $r(\cdot,\cdot,\cdot)$ such that when given ε, δ and h as inputs, $\mathcal{L}_3$ runs in time bounded by* $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$*, and outputs a list of probability models $\mathcal{T}$ such that with probability at least $1-\delta$ there exists some $\hat{T} \in \mathcal{T}$ such that $err(\hat{T}) \leq \varepsilon$, as long as the hypothesis h satisfies $err(h) \leq 1/r(1/\varepsilon, 1/\delta, k)$.*

Proof Our algorithm will first call the oracle Gen for $N = C\, m_2(2/\varepsilon, 4/\delta, k)\left(\frac{M^2}{\varepsilon^2}\log(1/\delta)\right)$ times, where C is some constant (to be determined in the following analysis) and m_2 is the polynomial upper bound for the runtime of the algorithm defined in Lemma 14. Then the algorithm will separate these data points (x, y)'s into two samples, one for $h(x) = 0$ and the other for $h(x) = 1$. For each sample corresponding to $h(x) = j$, if the sample size is at least $m = m_2(2/\varepsilon, 4/\delta)$, the run the learning algorithm $\mathcal{L}_2$ in Lemma 14 to the sample with target accuracy $\varepsilon/2$ and failure probability $\delta/4$ and obtain a polynomial list of distributions $\mathcal{P}_j$; otherwise, simply output a singleton list containing any arbitrary distribution in $\mathcal{P}$.

Let $j \in \{0, 1\}$ and $\pi_j = \Pr_{x\sim\mathcal{D}}[h(x) = j]$. Let us first consider the case where $\pi_j \geq \varepsilon/(2M)$. In order to invoke Lemma 32, we will upper bound the quantity $w_j \mathrm{KL}(P_j||\hat{P}_j)$, where $w_j = \Pr_{x\sim\mathcal{D}}[c(x) = j]$. We know that for some large enough constant C, we can guarantee with probability at least $1-\delta/4$, we will collect at least m observations with $h(x) = j$. Let $\varepsilon_h = err(h)$, note that when we instantiate the learner $\mathcal{L}_2$ on the sample with $h(x) = j$, the input distribution I_j is a $(\varepsilon_h, 1-\varepsilon_h)$-mixture of the distributions P_{1-j} and P_j. Then there exists a polynomial r such that if $err(h) \leq 1/r(1/\varepsilon, 1/\delta, k)$, we can have the following based on Lemma 32

$$\mathrm{KL}(P_j||I_j) \leq \varepsilon_h \mathrm{KL}(P||Q) \leq 1/m_{\mathcal{P}}(2/\varepsilon, 4/\delta, k)$$

where $m_{\mathcal{P}}$ is the polynomial defined in Lemma 14. This means, the learning algorithm $\mathcal{L}_2$ will with probability at least $1-\delta/4$, returns some distribution $\hat{P}_j$ in the output list such that $\mathrm{KL}(P_j||\hat{P}_j) \leq \varepsilon/2$, which implies that $w_j \mathrm{KL}(P_j||\hat{P}_j) \leq \varepsilon/2$.

Suppose that $\pi_j < \varepsilon/(2M)$, then we know that no matter what the distribution $\hat{P}_j$ is, we have $w_j \mathrm{KL}(P_j||\hat{P}_j) \leq \frac{\varepsilon}{2M} M = \varepsilon/2$ by Assumption 4.

Finally, our algorithm will output a list of probability models $\mathcal{T} = \{(h, \hat{P}_0, \hat{P}_1) \mid \hat{P}_0 \in \mathcal{P}_0, \hat{P}_1 \in \mathcal{P}_1\}$, such that with probability at least $1 - \delta$, there exists some model $\hat{T} = (h, \hat{P}_0, \hat{P}_1) \in \mathcal{T}$ such that

$$err(T) = w_0 \mathrm{KL}(P_0||\hat{P}_0) + w_1 \mathrm{KL}(P_1||\hat{P}_1) \leq \varepsilon,$$

which recovers our claim. ■

Lemma 34 *Suppose that the distribution class $\mathcal{P}$ is PAC learnable. Let P be the unconditional mixture distribution over the outcomes $\mathcal{Y}$ under the distribution Gen. Let $\varepsilon > 0$ and $\delta \in (0, 1)$. Then there exists an algorithm $\mathcal{L}_4$ and a polynomial $g(\cdot, \cdot, \cdot)$ such that when $\mathcal{L}_4$ is given sample access to Gen and ε, δ as inputs, it runs in time bounded by $\mathrm{poly}(1/\varepsilon, 1/\delta, k)$ and it will with probability at least $1 - \delta$, output a list of distributions $\mathcal{P}'$ that contains $\hat{P}$ with $\mathbb{E}_{x \sim \mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P})\right] \leq \varepsilon$, as long as P is an η-unhealthy mixture for some $\eta \leq 1/g(k, 1/\varepsilon, 1/\delta)$.*

Proof We first consider the case where the weight on one component is small, and without loss of generality assume that $w_1 \leq \varepsilon/(4M\,m)$. By Lemma 32 and Assumption 4, we know that

$$\mathrm{KL}(P_0||R) \leq w_1 \mathrm{KL}(P_0||P_1) \leq \frac{\varepsilon}{2M\,m} M \leq 1/(2m).$$

By instantiating the algorithm in Lemma 14 with parameters $(\varepsilon/2, \delta)$, we know with probability $1 - \delta$, there exists a hypothesis distribution $\hat{P}$ in the output list such that $\mathrm{KL}(P_0||\hat{P}) \leq \varepsilon/2$. Again by our Assumption 4, we know $\mathrm{KL}(P_1||\hat{P}) \leq M$, so it follows that

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P})\right] = w_0 \mathrm{KL}(P_0||\hat{P}) + w_1 \mathrm{KL}(P_1||\hat{P}) \leq \frac{\varepsilon}{2} + \frac{\varepsilon\,\mathrm{KL}(P_1||\hat{P})}{2M\,m} \leq \varepsilon.$$

Next suppose that we are in the second case where $\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0) \leq 1/(2m)$. We know from Lemma 32 that

$$\mathrm{KL}(P_0||R) \leq w_1 \mathrm{KL}(P_0||P_1) \leq 1/(2m) \quad \text{and,} \quad \mathrm{KL}(P_1||R) \leq w_0 \mathrm{KL}(P_1||P_0) \leq 1/(2m)$$

We will also apply the algorithm in Lemma 14 which guarantees with probability at least $1 - \delta$ that there exists a hypothesis distribution $\hat{P}$ in the output list $\mathcal{P}'$ such that $\mathrm{KL}(P_0||\hat{P}), \mathrm{KL}(P_1||\hat{P}) \leq \varepsilon/2$, which implies that

$$\mathop{\mathbb{E}}_{x \sim \mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P})\right] = w_0 \mathrm{KL}(P_0||\hat{P}) + w_1 \mathrm{KL}(P_1||\hat{P}) \leq \varepsilon.$$

Therefore, there exists a distribution $\hat{P}$ in the output list that satisfies our claim as long as we choose the polynomial g such that $g(1/\varepsilon, 1/\delta, k) \geq \max\{2Mm/\varepsilon, 2m\}$ for all ε, δ and m. ■

Proof of Theorem 10 We will now combine the all the tools to prove Theorem 10. First, consider the class of events $\mathcal{E}(\gamma)$ with $\gamma = 1/g(1/\varepsilon, 1/\delta, k)$ (specified in Lemma 17). Then we will apply the CN algorithm $\mathcal{L}_2$ in Lemma 11 to obtain a list H of polynomially many hypotheses. For each $h \in H$, run the algorithm $\mathcal{L}_3$ with h as a candidate hypothesis. This will generate a list of a list of probability models $\mathcal{T}$. If $\max\{\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0)\} \geq \gamma$, then $\mathcal{T}$ is guaranteed to contain an ε-accurate model with high probability (based on Lemma 11 and Lemma 15). Next, apply the distribution learner in Lemma 17 over the mixture distribution over $\mathcal{Y}$. If the algorithm outputs a distribution $\hat{P}$, create a model $T' = (h_0, \hat{P}, \hat{P})$, where hypothesis h_0 labels every example as negative. If $\max\{\mathrm{KL}(P_0||P_1), \mathrm{KL}(P_1||P_0)\} < \gamma$, we know T' is ε-accurate with high probability (based on Lemma 17). Finally, apply the *maximum likelihood* method to the list of models $\mathcal{T} \cup \{T'\}$: draw a sample of polynomial size from Gen, then for each model $T \in \mathcal{T} \cup \{T'\}$, compute the empirical log-loss over the sample, and output the model with the minimum log loss. By standard argument, we can show that the output model is accurate with high probability.

A.4. Missing Proofs in Section 5

Lemma 35 *Fix any $\gamma \in (0, 1]$. Suppose that $\mathrm{KL}(P||Q) \geq \gamma$, then $E(P, Q, \gamma/2)$ is a $(\gamma^2/(8M))$-distinguishing event for the distributions P and Q.*

Proof Note that for any $y \in E$ such that $P(E) > 0$, we have $\log \frac{P(y)}{Q(y)} \leq M$ by Assumption 4, and for any $y \notin E$, we also have $\log\left(\frac{P(y)}{Q(y)}\right) < \gamma/2$.

$$\begin{aligned}
\mathrm{KL}(P||Q) &= \int_{y\in\mathcal{Y}} P(y) \log \frac{P(y)}{Q(y)} dy \\
&= \int_{y\in E} P(y) \log \frac{P(y)}{Q(y)} dy + \int_{y\notin E} P(y) \log \frac{P(y)}{Q(y)} dy \\
&< P(E)M + (1 - P(E))\frac{\gamma}{2} \\
&= \frac{\gamma}{2} + (M - \gamma/2)P(E) < \frac{\gamma}{2} + M\,P(E)
\end{aligned}$$

Since we know that $\mathrm{KL}(P||Q) \geq \gamma$, it follows that $P(E) > \frac{\gamma}{2M}$. Furthermore,

$$\begin{aligned}
P(E) - Q(E) &= P(E)\left(1 - \frac{Q(E)}{P(E)}\right) \\
&\geq P(E)\left(1 - \sup_{y\in E} \frac{Q(y)}{P(y)}\right) \\
&\geq P(E)\left(1 - 2^{-\gamma/2}\right) \geq \frac{\gamma\,P(E)}{4}
\end{aligned}$$

where the last step follows from the fact that $1 - 2^{-a} \geq a/2$ for any $a \in [0, 1]$. It follows that

$$P(E) - Q(E) > \frac{\gamma\,P(E)}{4} > \frac{\gamma}{2M}\frac{\gamma}{4} = \frac{\gamma^2}{8M},$$

which proves our statement. ■

Lemma 36 *Suppose that the distributions $P, \hat{P}, Q, \hat{Q}$ over $\mathcal{Y}$ satisfy that $\mathrm{KL}(P||\hat{P}) \leq \alpha$, $\mathrm{KL}(Q||\hat{Q}) \leq \alpha$, and $\mathrm{KL}(P||Q) \geq \gamma$ for some $\alpha, \gamma \in (0, 1]$. Then the event $E(\hat{P}, \hat{Q}, (\gamma^2/(8M) - \sqrt{2\alpha})^2)$ is a ξ-distinguishing event with $\xi \geq 1/\operatorname{poly}(1/\gamma, 1/\alpha, k)$ as long as $\gamma > 8M(\sqrt{2\alpha} + (8M^2\alpha)^{1/8})$.*

Proof Since we have both $\mathrm{KL}(P||\hat{P}), \mathrm{KL}(Q||\hat{Q}) \leq \alpha$, by Pinsker's inequality, we can bound the total variation distances

$$\|P - \hat{P}\|_{tv} \leq \sqrt{\alpha/2} \quad \text{and,} \quad \|Q - \hat{Q}\|_{tv} \leq \sqrt{\alpha/2}.$$

By Lemma 20 and the definition of total variation distance, we know that

$$\|P - Q\|_{tv} = \sup_{E \subset \mathcal{Y}} |P(E) - Q(E)| \geq \gamma^2/(8M)$$

By triangle inequality, the above implies

$$\|\hat{P} - \hat{Q}\|_{tv} \geq \frac{\gamma^2}{8M} - \sqrt{2\alpha} \equiv b$$

By Pinsker's inequality, we know that $\|\hat{P} - \hat{Q}\|_{tv} \leq \sqrt{\mathrm{KL}(\hat{P}||\hat{Q})/2}$. It follows that $\mathrm{KL}(\hat{P}||\hat{Q}) \geq 2b^2$. Consider the event $E = E(\hat{P}, \hat{Q}, b^2)$. We know by Lemma 20 that E is a $(b^4/(2M))$-distinguishing event for distributions $\hat{P}$ and $\hat{Q}$. Since both $\mathrm{KL}(P||\hat{P}), \mathrm{KL}(Q||\hat{Q}) \leq \alpha$, we have

$$|P(E) - \hat{P}(E)| \leq \|P(E') - \hat{P}(E')\|_{tv} \leq \sqrt{\alpha/2} \quad \text{and,} \quad |Q(E) - \hat{Q}(E)| \leq \|Q(E') - \hat{P}(E')\|_{tv} \leq \sqrt{\alpha/2}.$$

Since E is a $(b^4/(2M))$-distinguishing event for the distributions $\hat{P}$ and $\hat{Q}$, this means $|\hat{P}(E) - \hat{Q}(E)| \geq (b^4/(2M))$, and by triangle inequality, we have

$$\begin{aligned} |P(E) - Q(E)| &= |(P(E) - \hat{P}(E)) + (\hat{P}(E) - \hat{Q}(E)) + (\hat{Q}(E) - Q(E))| \\ &\geq |\hat{P}(E) - \hat{Q}(E)| - |P(E) - \hat{P}(E)| - |\hat{Q}(E) - Q(E)| \\ &\geq (b^4/(2M)) - \sqrt{2\alpha} \end{aligned}$$

Note that if we have $\gamma > 8M(\sqrt{2\alpha} + (8M^2\alpha)^{1/8})$, then we can guarantee both $b > 0$ and $(b^4/(2M)) - \sqrt{2\alpha} > 0$. ■

Lemma 37 *Suppose the class $\mathcal{P}$ satisfies the parametric mixture learning assumption (Assumption 18), the class $\mathcal{C}$ is CN learnable, and mixture distribution over $\mathcal{Y}$ is γ-healthy for some $\gamma > 0$. Then there exists an algorithm $\mathcal{L}$ that given ε, δ and γ as inputs and sample access from Gen, halts in time bounded by $\operatorname{poly}(1/\varepsilon, 1/\delta, 1/\gamma, n, k)$, and with probability at least $1 - \delta$, outputs a list of probability models $\mathcal{T}$ that contains some $\hat{T}$ with $err(\hat{T}) \leq \varepsilon$.*

Proof We will first invoke the algorithm $\mathcal{L}_M$ in Assumption 18 so that with probability at least $1 - \delta/2$, the output approximations for the two components satisfy $\mathrm{KL}(P_0||\hat{P}_0) \leq \alpha$ and $\mathrm{KL}(P_1||\hat{P}_1) \leq \alpha$ for some α that satisfies $\gamma > 8M(\sqrt{2\alpha} + (8M^2\alpha)^{1/8})$. This process will halt in time $\operatorname{poly}(1/\alpha, 1/\delta, 1/\gamma, k)$.

By Lemma 20, we know that the either event $E(\hat{P}_0, \hat{P}_1, \gamma/2)$ is a ξ-distinguishing event for P_0 and P_1 for some $\xi \geq 1/\operatorname{poly}(1/\gamma, n, k)$. Then we can use the CN learning algorithm $\mathcal{L}_1$ in

Lemma 8 with the distinguishing event E to learn a list of hypotheses H under polynomial time, and there exists some $h \in H$ that is ε_1 accurate, with $\varepsilon_1 = 1/r(1/\varepsilon, 1/\delta, k)$ (specified in Lemma 15). For each hypothesis $h' \in H$, run the algorithm $\mathcal{L}_3$ with h' as the candidate hypothesis and ε as the target accuracy parameter. By Lemma 15, this will halt in polynomial time, and outputs a list of probability models $\mathcal{T}$ such that one of which has error $err(\hat{T}) \leq \varepsilon$. ■

Proof of Theorem 19 The algorithm consists of three steps. First, we will run the algorithm in Lemma 22 by setting $\gamma = 1/g(1/\varepsilon, \delta, k)$ (specified in Lemma 15) and other parameters in a way to guarantee that whenever $\max\{\mathrm{KL}(P_0||\hat{P}_0), \mathrm{KL}(P_1||\hat{P}_1)\} \geq \gamma$ and $\min\{w_0, w_1\} \geq \gamma$ both hold, the output list of models $\mathcal{T}$ contains some T that has error at most ε. Next, we will directly apply the distribution learner in Lemma 17 so that when the healthy mixture condition is not met, the algorithm outputs a distribution $\hat{P}$ such that $\mathbb{E}_{x\sim\mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P})\right]$. Lastly, similar to the final step in the forward reduction, we run the *maximum likelihood* algorithm to output the model in $\mathcal{T} \cup \{(h_0, \hat{P}, \hat{P})\}$ with the smallest empirical log-loss.

Appendix B. Maximum Likelihood Algorithm

In this section, we will formally define the maximum likelihood algorithm, which is a useful subroutine to select an accurate probability model from a list of candidate models. First, to give some intuition, we show that the objective of minimizing $\mathbb{E}_{x\sim\mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P}_{h(x)})\right]$ is equivalent to minimizing the expected log-losses. For any distribution $\hat{P}$ over $\mathcal{Y}$ and a point $r \in \mathcal{Y}$, the *log likelihood* loss (or simply log-loss) is defined as $\mathrm{loss}(y, \hat{P}) = -\log \hat{P}(y)$. The *entropy* of a distribution P over range $\mathcal{Y}$, denoted $H(P)$, is defined as

$$H(P) = \int_{y\in\mathcal{Y}} P(y) \log \frac{1}{P(y)} dy$$

For any two distributions P and $\hat{P}$ over $\mathcal{Y}$, we could write KL-divergence as

$$\mathrm{KL}(P||\hat{P}) = \int_{y\in\mathcal{Y}} P(y) \log \frac{1}{\hat{P}(y)} dy - H(P) = \mathop{\mathbb{E}}_{y\sim P}\left[-\log \hat{P}(y)\right] - H(P) \tag{2}$$

which will be useful for proving the next lemma.

Lemma 38 *Given any hypothesis $h\colon \mathcal{X} \to \{0,1\}$, and hypothesis distributions $\hat{P}_0$ and $\hat{P}_1$, we have*

$$\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||\hat{P}_{h(x)})\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right] - \mathop{\mathbb{E}}_{(x,y)\sim \mathit{Gen}}\left[\log(\hat{P}_{h(x)}(y))\right]$$

Proof We can write the following

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\mathrm{KL}(P_{c(x)}||P_{h(x)})\right] &= \Pr_{\mathcal{D}}[c(x)=1, h(x)=1]\,\mathrm{KL}(P_1||\hat{P}_1) + \Pr_{\mathcal{D}}[c(x)=1, h(x)=0]\,\mathrm{KL}(P_1||\hat{P}_0) \\
&\quad + \Pr_{\mathcal{D}}[c(x)=0, h(x)=1]\,\mathrm{KL}(P_0||\hat{P}_1) + \Pr_{\mathcal{D}}[c(x)=0, h(x)=0]\,\mathrm{KL}(P_0||\hat{P}_0) \\
\text{(apply Equation (2))} \quad &= \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right] - \sum_{(i,j)\in\{0,1\}^2} \Pr_{\mathcal{D}}[c(x)=i, h(x)=j] \mathop{\mathbb{E}}_{y\sim P_i}\left[\log(\hat{P}_j(y))\right] \\
&= \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right] - \mathop{\mathbb{E}}_{(x,y)\sim\mathrm{Gen}}\left[\log(\hat{P}_{h(x)}(y))\right]
\end{aligned}$$

which proves our claim. ■

Therefore, we could write $err(T) = \mathbb{E}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right] - \mathbb{E}_{(x,y)\sim\text{Gen}}\left[\log(\hat{P}_{h(x)}(y))\right]$ for any model $T = (h, \hat{P}_0, \hat{P}_1)$. Observe that $\mathbb{E}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right]$ is independent of the choices of $(h, \hat{P}_0, \hat{P}_1)$, so our goal can also be formulated as minimizing the expected log-loss $\mathbb{E}_{(x,y)\sim\text{Gen}}\left[\log(\hat{P}_{h(x)}(y))\right]$. To do that, we will use the following *maximum likelihood* algorithm: given a list of probability models $\mathcal{T}$ as input, draw a set of S of samples (x, y)'s from Gen, and for each $T = (h, \hat{P}_0, \hat{P}_1) \in \mathcal{T}$, compute the log-loss on the sample

$$\text{loss}(S, T) = \sum_{(x,y)\in S} \text{loss}(y, P_{h(x)}),$$

and lastly output the probability model $\hat{T} \in \mathcal{T}$ with the smallest $\text{loss}(S, T)$.

Our goal is to show that if the list of models $\mathcal{T}$ contains an accurate model T, the maximum likelihood algorithm will then output an accurate model with high probability.

Theorem 39 *Let $\varepsilon > 0$. Let $\mathcal{T}$ be a set of probability models such that at least one model $T^* \in \mathcal{T}$ has error $err(T^*) \leq \varepsilon$. Suppose that the class $\mathcal{P}$ also satisfies bounded assumption (in Assumption 4).*

If we run the maximum likelihood algorithm on the list $\mathcal{T}$ using a set S of independent samples drawn from Gen. Then, with probability at least $1 - \delta$, the algorithm outputs some model $\hat{T} \in \mathcal{T}$ such that $err(\hat{T}) \leq 4\varepsilon$ with

$$\delta \leq (|\mathcal{T}| + 1) \exp\left(\frac{-2m\varepsilon^2}{M^2}\right).$$

To prove this result, we rely on the Hoeffding concentration bound.

Theorem 40 *Let $x_1, \ldots, x_n$ be independent bounded random variables such that each x_i falls into the interval $[a, b]$ almost surely. Let $X = \sum_i x_i$. Then for any $t > 0$ we have*

$$\Pr[X - \mathbb{E}[X] \geq t] \leq \exp\left(\frac{-2t^2}{n(b-a)^2}\right) \quad \textit{and} \quad \Pr[X - \mathbb{E}[X] \leq -t] \leq \exp\left(\frac{-2t^2}{n(b-a)^2}\right)$$

Proof Our proof essentially follows from the same analysis of Feldman et al. (2008) (Theorem 17). We say that a probability model T is *good* if $err(T) \leq 4\varepsilon$, and *bad* otherwise. We know that $\mathcal{T}$ is guaranteed to contain at least one good model. In the following, we will write $H(\text{Gen})$ to denote $\mathbb{E}_{x\sim\mathcal{D}}\left[H(P_{c(x)})\right]$.

The probability δ that the algorithm fails to output some good model is at most the probability the best model T^* has $\text{loss}(S, T) \geq m\left(H(\text{Gen}) + 2\varepsilon\right)$ or some bad model T' has $\text{loss}(S, T') \leq m\left(H(\text{Gen}) + 3\varepsilon\right)$. Applying union bound, we get

$$\delta \leq |\mathcal{T}| \Pr[\text{loss}(S, T') \leq m\left(H(\text{Gen}) + 3\varepsilon\right) \mid err(T) \geq 4\varepsilon] + \Pr[\text{loss}(S, T^*) \geq m\left(H(\text{Gen}) + 2\varepsilon\right)]$$

For each bad model T' with $err(T') > 4\varepsilon$, we can write

$$\begin{aligned}
\Pr[\text{loss}(S,T') \le m(H(\text{Gen}) + 3\varepsilon)] &= \Pr[\text{loss}(S,T') \le m(H(\text{Gen}) + 4\varepsilon) - \varepsilon m] \\
\text{(because } err(T') \ge 0) \qquad &\le \Pr[\text{loss}(S,T') \le m(H(\text{Gen}) + err(T')) - \varepsilon m] \\
&= \Pr[\text{loss}(S,T') \le \mathop{\mathbb{E}}_{S\sim\text{Gen}^m}\left[\text{loss}(S,T') - \varepsilon\right]] \\
&\le \exp\left(\frac{-2m\varepsilon^2}{M^2}\right)
\end{aligned}$$

where the last step follows from Theorem 40. Similarly, for the best model T^* with $err(T^*) \le \varepsilon$, we have the following derivation:

$$\begin{aligned}
\Pr[\text{loss}(S,T^*) \ge m\left(H(\text{Gen}) + 2\varepsilon\right)] &= \Pr[\text{loss}(S,T^*) \ge m\left(H(\text{Gen}) + \varepsilon\right) + m\varepsilon] \\
&\le \Pr[\text{loss}(S,T^*) \ge m\left(H(\text{Gen}) + err(T^*) + m\varepsilon\right)] \\
&= \Pr[\text{loss}(S,T^*) \ge \mathop{\mathbb{E}}_{S\sim\text{Gen}^m}\left[\text{loss}(S,T^*)\right] + m\varepsilon] \\
&\le \exp\left(\frac{-2m\varepsilon^2}{M^2}\right)
\end{aligned}$$

Combining these two probabilities recovers the stated bound. ■

In other words, as long as we have an ε-accurate model in the list, we can guarantee with probability at least $1-\delta$ that the output model has error $O(\varepsilon)$ using a sample of size no more than $\text{poly}(k/\varepsilon) \cdot \log(1/\delta)$.

Appendix C. Examples of Distinguishing Events

In this section, we give two distribution classes that admit distinguishing event class of polynomial size.

C.1. Spherical Gaussian

We consider the class of spherical Gaussian in $\mathbb{R}^k$ with fixed covariance and bounded means. In particular, let

$$\mathcal{P} = \{\mathcal{N}(\mu, \Sigma) \mid \mu \in [0,1]^k\}$$

where Σ is some diagonal covariance matrix in $\mathbb{R}^{k\times k}$ such that the variance in each coordinate satisfy $0 < \sigma_j^2 \le \sigma^2$ for some constant $\sigma > 1$.

Theorem 41 *There exists a parametric class of events $\mathcal{E}(\cdot)$ for the distribution class $\mathcal{P}$ of k-dimensional Spherical Gaussian such that for any $\gamma > 0$ and for any two probability distributions P and Q in the class $\mathcal{P}$ such that $\text{KL}(P||Q) \ge \gamma$, the class of events $\mathcal{E}(\gamma)$ contains an event E that is an ξ-distinguishing event, where $\max\{1/\xi, |\mathcal{E}(\gamma)|\} \le \text{poly}(k, 1/\gamma)$.*

Proof Recall that the KL divergence of two multivariate Gaussian distributions P and Q with means μ, μ' and covariance matrices Σ_p, Σ_q can be written as

$$\text{KL}(P||Q) = \frac{1}{2}\left(\text{tr}(\Sigma_q^{-1}\Sigma_p) + (\mu'-\mu)^\intercal \Sigma_q (\mu'-\mu) - k + \log\left(\frac{\det\Sigma_q}{\det\Sigma_p}\right)\right).$$

For any two distributions P and Q in our class $\mathcal{P}$, we can simplify the KL divergence as

$$\mathrm{KL}(P||Q) \leq \frac{\sigma^2}{2}\|\mu - \mu'\|_2^2.$$

Then $\mathrm{KL}(P||Q) \geq \gamma$ implies that there exists some coordinate $j \in [k]$ such that $|\mu_j - \mu'_j| \geq \sqrt{2\gamma/(k\sigma^2)}$. Note that the marginal distributions of P_j and Q_j over the j-the coordinate are $\mathcal{N}(\mu_j, \sigma_j^2)$ and $\mathcal{N}(\mu'_j, \sigma_j^2)$ respectively. Without loss of generality, assume that $\mu'_j < \mu_j$. Then for any value $t \in [\mu'_j, \mu_j]$, we have

$$P_j[y \geq t] - Q_j[y \geq t] \geq P_j[y \in [t, \mu_j]]. \tag{3}$$

Let $\Delta = \sqrt{2\gamma/(k\sigma^2)}$, and consider the discretized set $L(\gamma) = \{0, \Delta, \ldots, \lfloor 1/\Delta \rfloor \Delta\}$. Then we know there exists a value $t' \in L$ such that $t' \in L(\gamma)$ such that $t' \in [\mu'_j, \mu_j]$ and $\mu_j - t' \geq \Delta$. By Equation (3), we can write

$$P_j[y \geq t'] - Q_j[y \geq t'] \geq \frac{1}{2}\mathrm{erf}(\Delta/(\sqrt{2}\sigma_j)) \geq \frac{1}{2}\mathrm{erf}(\Delta/(\sqrt{2}\sigma))$$

where erf denotes the Gauss error function with $\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-a^2}\,da$ for every $x \in \mathbb{R}$. The Taylor expansion of the function is

$$\mathrm{erf}(x) = \frac{2}{\sqrt{\pi}}\sum_{i=0}^{\infty}\frac{(-1)^i x^{2i+1}}{n!(2i+1)} = \frac{2}{\sqrt{\pi}}\left(x - \frac{x^3}{3} + \frac{x^5}{10} - \frac{x^7}{42}\cdots\right)$$

Therefore, for any $x \in [0, 1)$, there exists a constant C such that $\mathrm{erf}(x/(\sqrt{2}\sigma))/2 \geq C\,x$. It follows that

$$P_j[y \geq t'] - Q_j[y \geq t'] \geq C\Delta.$$

This means that the event of $(y_j \geq t')$ is a $(C\Delta)$-distinguishing event for the two distributions P and Q. Therefore, for any $\gamma > 0$, we can construct the following class of distinguishing events

$$\mathcal{E}(\gamma) = \{\mathbf{1}[y_j \geq t'] \mid j \in [k], t' \in L(\gamma)\}.$$

Note that both $1/(C\Delta)$ and $|\mathcal{E}(\gamma)|$ is upper bounded by $\mathrm{poly}(1/\gamma, k)$, which recovers our claim. ■

C.2. Product Distributions over Discrete Domains

Consider the space of b-ary cube $\mathcal{Y} = \{0, \ldots, b-1\}^k$, and the class of full-support product distributions $\mathcal{P}$ over $\mathcal{Y}$: distributions whose k coordinates are mutually independent distributions over $\{0, \ldots, b-1\}$. In particular, we assume that there exists some quantity $M \leq \mathrm{poly}(k, b)$ such that for each $P \in \mathcal{P}$ and each coordinate j and $y_j \in \{0, 1, \ldots b-1\}$, we have $\log(1/P_j(y_j)) \leq M$. Now let's show that this class of distributions admits a small class of distinguishing events as well.

Theorem 42 *There exists a parametric class of events $\mathcal{E}(\cdot)$ for the production distribution class over the b-ary cube such that for any $\gamma > 0$ and for any two probability distributions P and Q in the class $\mathcal{P}$ such that $\mathrm{KL}(P||Q) \geq \gamma$, the class of events $\mathcal{E}(\gamma)$ contains an event E that is an ξ-distinguishing event, where $\max\{1/\xi, |\mathcal{E}(\gamma)|\} \leq \mathrm{poly}(k, b, 1/\gamma)$.*

Proof In the following, we will write $P = P_1 \times \ldots \times P_k$ and $Q = Q_1 \times \ldots \times Q_k$. Note that

$$\mathrm{KL}(P||Q) = \sum_{j' \in [k]} \mathrm{KL}(P_{j'}||Q_{j'}).$$

Therefore $\mathrm{KL}(P||Q) \geq \gamma$ implies that there exists some coordinate j such that $\mathrm{KL}(P_j||Q_j) \geq \gamma/k$. This means

$$\sum_{y'_j \in \{0,\ldots,b-1\}} P_j(y'_j) \log\left(\frac{P_j(y'_j)}{Q_j(y'_j)}\right) \geq \gamma/k.$$

This means there exists some $t \in \{0, \ldots, b-1\}$ such that $P_j(t)\log(P_j(t)/Q_j(t)) \geq \gamma/(kb)$. Recall that $\log\left(P_j(t)/Q_j(t)\right) \leq M$, then we must have $P_j(t) \geq \gamma/(kbM)$. Furthermore, since $P_j(t) \leq 1$, we must also have $\log(P_j(t)/Q_j(t)) \geq \gamma/(kb)$. It follows that

$$P_j(t) - Q_j(t) \geq P_j(t)\left(1 - \frac{Q_j(t)}{P_j(t)}\right) \geq \frac{\gamma}{kbM}\left(1 - 2^{-\gamma/(kb)}\right) \geq \frac{\gamma}{kbM}\frac{\gamma}{2kb} = \frac{\gamma^2}{2(kb)^2 M}$$

where the last inequality follows from the fact that $1 - 2^{-z} \geq z/2$ for any $z \in [0, 1]$. Therefore, for any $\gamma > 0$, the following class of events

$$\mathcal{E}(\gamma) = \{\mathbf{1}[y_j = t] \mid t \in \{0, 1, \ldots, b-1\}, j \in [k]\}$$

would contain a ξ-distinguishing event, and $\max\{1/\xi, |\mathcal{E}(\gamma)|\} \leq \mathrm{poly}(k, b, 1/\gamma)$. ■