Predicting with Distributions

Michael Kearns* Zhiwei Steven Wu†

June 7, 2016

Abstract

We consider a new PAC-style 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 that partially resolves an open problem [RDM06], and an application of the Neyman-Pearson Lemma to obtain robust learning algorithms.

1 Introduction

We consider a new PAC-style 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 \(y\). This framework significantly generalizes settings such as learning with classification noise or errors (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 structured prediction (where \(y\) is a complex multi-dimensional object but still deterministic).

As in the standard PAC model, we begin with an unknown binary\(^1\) function or concept \(c\) chosen from a known class \(C\), 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 \(P\). 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 of learning is to minimize the conditional Kullback-Leibler (KL) divergence given \(c(x)\), rather than simply the KL divergence to the mixture. We thus consider problems of Predicting with Distributions (PwD) over large output spaces.

Our main interest is in conditions permitting computationally efficient learning in our model. Our main results take the form of reductions from our model to algorithms for PAC learning the concept class \(C\) and the distribution class \(P\) separately. Informally, our results imply that for virtually every concept class \(C\) known to be PAC learnable with classification noise, and every

\(^1\)We leave the consideration of multi- or real-valued functions \(c(x)\) to future work.
class $\mathcal{P}$ known to be PAC learnable in the distributional sense of [KMR+94], PwD problems given by $(\mathcal{C}, \mathcal{P})$ are learnable in our framework.

A centerpiece of our proofs 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 significantly different under $P_0$ and $P_1$. Provided these distributions are themselves sufficiently different. Our first main 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 $c$. As a consequence this construction answers an open problem posed in [RDM06] regarding PAC learning with noise rates exceeding $1/2$.

We then use distinguishing events to provide two different reductions of our model to extant PAC variants. In our “forward” reduction, we assume the distribution class $\mathcal{P}$ admits a small set of candidate distinguishing events. 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 $\mathcal{P}$. Thus, if $\mathcal{C}$ is PAC learnable with classification noise, $\mathcal{P}$ has a small set of distinguishing events, and $\mathcal{P}$ is PAC learnable, then $(\mathcal{C}, \mathcal{P})$ is learnable in our model.

In our “reverse” reduction, we instead first separate the distributions, then use their approximations to learn $c$. Here we need a stronger distribution-learning assumption, but not a small set of distinguishing events. More precisely, we assume that *mixtures* of two distributions from $\mathcal{P}$ (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 construct a specialized distinguishing event, which in turn lets us create a noisy label for $c$. Thus, if $\mathcal{P}$-mixtures are PAC learnable, and $\mathcal{C}$ is PAC learnable with classification noise, then $(\mathcal{C}, \mathcal{P})$ is learnable in our model.

In both reductions, we make central use of the Neyman-Pearson Lemma 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 [Kea98], with parity-based constructions being the only known exceptions), and the distributional 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 all known combinations.

### 1.1 Related Works

At the highest level, our model falls under the framework of [Hau92], which gives a decision-theoretic treatment of PAC-style learning [Val84] 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 [Hau92] is primarily concerned with sample complexity, our focus here is on computational complexity.

At a more technical level, our results make heavy use of results for PAC learning under classification noise [AL87, RDM06, Dec97, Kea98], and for PAC models for distribution learning [KMR+94, FSO06, Das99, FOS08].
2 Preliminaries

2.1 Model

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

Let \( C \) be a class of \( \{0,1\} \)-valued functions (also called concepts) over the context space \( X \), and \( P \) be a class of probability distributions over the outcome space \( Y \). Let \( D \) be an unknown target distribution over \( X \). Fix any target concept \( c \in C \) and target distributions \( P_0 \) and \( P_1 \) in \( P \), consider the following generative oracle \( \text{Gen}(D,c,P_0,P_1) \) (or simply Gen) that on each call does the following (see Figure 1 for an illustration):

- Draws a context \( x \) randomly and independently according to \( D \);
- Evaluates the concept \( c \) on \( x \), and draws an outcome \( y \) randomly from \( P_c(x) \);
- Returns the context-outcome pair \( (x,y) \).

A learning algorithm is given sample access to Gen and returns a probability model defined by a triple \( T = (h,\hat{P}_0,\hat{P}_1) \in C \times P \times P \).

Given any context \( x \), the model predicts the outcome \( y \) to be drawn from the distribution \( \hat{P}_{h(x)} \). We will measure the error of such models using Kullback-Leibler divergence (KL divergence):

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

Let \( w_0 = \Pr_{x \sim D} [c(x) = 0] \) and \( w_1 = \Pr_{x \sim D} [c(x) = 0] \) be the fractions of negative and positive examples for the target concept \( c \). The distribution over \( y \) is a mixture of distributions \( P_0 \) and \( P_1 \) with weights \( w_0 \) and \( w_1 \), and the target concept \( c \) determines which component the outcome \( y \) is drawn from. Note that our learning objective is more demanding than just learning the mixture distribution — we need to minimize the expected conditional KL divergence.

We now formally introduce our learning model termed as Predicting with Distributions learning (PwD-learning).

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

We note we have stated the definition for the “proper” learning case in which the hypothesis models lie in the target classes \( C \) and \( P \). However, all of our results hold for the more general case in which they lie in potentially richer classes \( C' \) and \( P' \).
Figure 1: The generative model of Gen with target distribution $D$, concept $c$ over $X$ and distributions $P_0, P_1$ over $Y$.

2.2 The Classification Noise Model

**Definition 2** (CN Learnability [AL87]). Let $C$ be a concept class over $X$. We say that $C$ is efficiently learnable with noise (CN learnable) if there exists a learning algorithm $L$ such that for any $c \in C$, any distribution $D$ over $X$, any noise rate $0 \leq \eta < 1/2$, and for any $0 < \epsilon \leq 1$ and $0 < \delta \leq 1$, the following holds: if $L$ is given inputs $\eta_b$ (where $1/2 > \eta_b \geq \eta$), $\epsilon, \delta, n$, and is given access to $EX^\eta_{CN}(c, D)$, then $L$ will halt in time bounded by $\text{poly}(1/(1 - 2\eta_b), 1/\epsilon, 1/\delta, n)$ and output a hypothesis $h \in C$ that with probability at least $1 - \delta$ satisfies $\text{err}(h) \leq \epsilon$.

**CCCN Learning** In a more general noise model called Class-Conditional Classification Noise (CCCN) proposed by [RDM06], the example oracle $EX^\eta_{CCCN}$ has class-dependent noise rates — that is, the noise rate $\eta_0$ for the negative examples $(c(x) = 0)$ and the noise rate $\eta_1$ for the positive examples $(c(x) = 1)$ may be different, but both below 1/2. Moreover, [RDM06] shows that any class that is learnable under CN is also learnable under CCCN. We will show that the constraint that both noise rates $\eta_0, \eta_1 < 1/2$ can be relaxed.

**Lemma 1** ([RDM06]). Suppose that the concept class $C$ is CN learnable. Then there exists an algorithm $L_C$ and a polynomial $m_C(\cdot, \cdot, \cdot, \cdot)$ such that for every target concept $c \in C$, any $\epsilon, \delta \in (0, 1)$, for any noise rates $\eta_0, \eta_1 \leq \eta_b < 1/2$, if $L$ is given inputs $\epsilon, \delta, \eta_b$ and access to $EX^\eta_{CCCN}(c, D)$, then $L$ will halt in time bounded by $m_C(1/(1 - 2\eta_b), 1/\epsilon, 1/\delta, n)$, and output with probability at least $1 - \delta$ a hypothesis $h$ with error $\text{err}(h) \leq \epsilon$. We will say that $L_C$ is an (efficient) CCCN learner for $C$ with sample complexity $m_C$.

2.3 Distribution Learning

We will now give a definition for learnable distribution classes. Since we are concerned with computational complexity in our model, we need to specify how a distribution is represented.

Throughout the paper, we assume that we have access to any distribution in class $P$ through an evaluator, which allows us to evaluate the probability (density) at any point $y$ assigned by any distribution $P \in P$.

**Definition 3** (Evaluator [KMR+94]). Let $P$ be a class of distributions over the outcome space $Y$. We say that $P$ has an efficient evaluator if there exists a polynomial $p$ such that for any $n \geq 1$, and for any distribution $P \in P$, there exists an algorithm $E_P$ with runtime bounded by $\text{poly}(k)$ that given an input $y \in Y$ outputs the probability (density) assigned to $y$ by $P$. Thus, if $y \in Y$, then $E_P(y)$ is the weight of $y$ under $P$. We call $E_P$ an evaluator for $P$. 
The technique introduced here will partially answer this question. However, let us consider an example in which 
$$P(x, y)$$ is more likely to occur under $$P$$ if one noise rate is above 1/2. We can simply flip the labels to make both rates below 1/2.

In this section, we will introduce a central concept to our framework—distinguishing events. Informally, an event $$E \subseteq \mathcal{Y}$$ is distinguishing if it occurs with different probabilities under the measures $$P_0$$ and $$P_1$$, and therefore conveys information about the target concept $$c$$. We will rely on such events to create CN learning problems, which is crucial for both our forward and reverse reductions in subsequent sections.

**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 $$\xi$$-distinguishing for distributions $$P$$ and $$Q$$ if $$|P(E) - Q(E)| \geq \xi$$. We will call $$\xi$$ the separation parameter for such an event.

We show that the knowledge of a distinguishing event between $$P_0$$ and $$P_1$$ allows us to simulate an example oracle $$EX^n_{CCCN}$$ and therefore we can learn the concept $$c$$ with a CCCN learner. The main technical challenge is to ensure that noise rates $$\eta_0$$ and $$\eta_1$$ of the oracle are strictly less than 1/2. To see the issue, consider an example in which $$P_1(E) = 1/3$$ and $$P_0(E) = 1/4$$. Since the event is more likely to occur under $$P_1$$, we might assign the label 1 to an example whenever the event occurs and 0 otherwise, but then the noise rates are $$\eta_0 = 1/4$$ and $$\eta_1 = 2/3$$. This is related to a question left open by [RDM06]: Is it possible to learn when one noise rate is above 1/2 and the other is below 1/2? The technique introduced here will partially answer this question.

Our solution is to construct a randomized mapping from the event to the labels. Let us first introduce some parameters. Let $$E \subseteq \mathcal{Y}$$ be a $$\xi$$-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 $$\text{Lab}(\hat{p}, \hat{q}, \xi)$$ that takes parameters $$\hat{p}, \hat{q}$$ that are estimates for $$p$$ and $$q$$, and the separation parameter $$\xi$$ as inputs, and outputs a labeled example $$(x, \ell)$$ as follows:

- Draw an example $$(x, y)$$ from the oracle $$\text{Gen}$$.

\[\text{Lab}(\hat{p}, \hat{q}, \xi) = \begin{cases} \text{label } x \text{ with probability } \hat{p}, \\ \text{label } x \text{ with probability } \hat{q}. \end{cases}\]

\[\text{Label}(\hat{p}, \hat{q}, \xi) = \frac{\max\{\hat{p}, \hat{q}\} + \xi}{2}\]

\[\text{Label}(\hat{p}, \hat{q}, \xi) = \frac{\min\{\hat{p}, \hat{q}\} - \xi}{2}\]

\[\text{If } \hat{p} > \xi \text{ and } \hat{q} > \xi, \text{ then label } x \text{ with probability } \hat{p}, \text{ and label } x \text{ with probability } \hat{q}.\]

\[\text{If } \hat{p} > \xi \text{ and } \hat{q} < \xi, \text{ then label } x \text{ with probability } \hat{p}, \text{ and label } x \text{ with probability } \hat{q}.\]

\[\text{If } \hat{p} < \xi \text{ and } \hat{q} > \xi, \text{ then label } x \text{ with probability } \hat{p}, \text{ and label } x \text{ with probability } \hat{q}.\]

\[\text{If } \hat{p} < \xi \text{ and } \hat{q} < \xi, \text{ then label } x \text{ with probability } \hat{p}, \text{ and label } x \text{ with probability } \hat{q}.\]

Due to this randomized mapping, the example oracle $$EX^n_{CCCN}$$ can be simulated with 
$$n = \frac{1}{2\xi} \left\lceil \frac{1}{2\xi} \right\rceil \log \frac{1}{\xi}.$$
• 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

\[
\begin{align*}
    a_0 &= 1/2 + \frac{\xi(\hat{p} + \hat{q} - 2)}{4(\hat{q} - \hat{p})} \\
    a_1 &= 1/2 - \frac{\xi(\hat{p} + \hat{q} - 2)}{4(\hat{q} - \hat{p})} \\
    b_0 &= 1/2 + \frac{\xi(\hat{p} + \hat{q})}{4(\hat{q} - \hat{p})} \\
    b_1 &= 1/2 - \frac{\xi(\hat{p} + \hat{q})}{4(\hat{q} - \hat{p})}
\end{align*}
\]

(1)

(2)

• Output the labeled example \((x, \ell)\).

It’s easy to check that both vectors \((a_0, a_1)\) and \((b_0, b_1)\) form valid probabilities over \([0,1]\).

**Claim 1.** The values of \(a_0, a_1, b_0, b_1\) satisfy \(a_0 + a_1 = b_0 + b_1 = 1\) and \(a_0, b_0 \in [0,1]\).

**Proof.** It is easy to check that in Equations (1) and (2) the equations, we must have \(a_0 + a_1 = b_0 + b_1 = 1\). 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. \(\square\)

We will now show how we can ensure the noise rates to be below 1/2. As a first step, we give expressions for the noise rates of \textbf{Lab} in terms of the true probabilities \(p\) and \(q\), and show that the “estimated” noise rates in terms of \(\hat{p}\) and \(\hat{q}\) are below \((1/2 - \xi/4)\).

**Lemma 2.** Given a fixed distinguishing event \(E\), the rate of mislabeling positive and negative examples by \textbf{Lab} are

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

Moreover, given any input estimates \((\hat{p}, \hat{q})\) for \((p, q)\), the estimated noise rates 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

\[
\Pr[\ell = 0 \mid c(x) = 1] = \Pr[(\ell = 0) \wedge (y \in E) \mid c(x) = 1] + \Pr[(\ell = 0) \wedge (y \notin E) \mid c(x) = 1]
\]

\[
= \Pr_{\text{Gen}}[y \in E \mid c(x) = 1] \Pr_{\text{Lab}}[\ell = 0 \mid (y \in E) \wedge (c(x) = 1)]
\]

\[
+ \Pr_{\text{Gen}}[y \notin E \mid c(x) = 1] \Pr_{\text{Lab}}[\ell = 0 \mid (y \notin E) \wedge (c(x) = 1)]
\]

\[
= \Pr_{\text{Gen}}[y \in E \mid c(x) = 1]a_0 + \Pr_{\text{Gen}}[y \notin E \mid c(x) = 1]b_0
\]

\[
= qa_0 + (1-q)b_0
\]
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. \hfill \Box

To obtain accurate estimates $\hat{p}$ and $\hat{q}$, we will guess the values of $p$ and $q$ on a grid of size $[1/\Delta]^2$, where $\Delta \in [0,1]$ is some discretization parameter. Note that for some pair of values $i,j \in [1/\Delta]$ and $i \neq j$ such that the guesses $(\hat{p}, \hat{q}) = (i\Delta, j\Delta)$ satisfies

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

Now we will show that given such accurate guesses $\hat{p}$ and $\hat{q}$, we can use $\text{Lab}$ to simulate the example oracle $EX^{n}_{\text{CCCN}}$, and we derive the noise rates $\eta_0$ and $\eta_1$ below.

**Lemma 3.** Given any choice of discretization parameter $\Delta$, there exists some guess values $\hat{p}$ and $\hat{q}$ for $p$ and $q$ such that $|p - \hat{p}| \leq \Delta$ and $|q - \hat{q}| \leq \Delta$. Moreover, the noise rates of $\text{Lab}(\hat{p}, \hat{q}, \xi)$ for both the positive and negative examples are 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

$$\eta_1 - (\hat{q}a_0 + (1-\hat{q})b_0) = (qa_0 + (1-q)b_0) - (\hat{q}a_0 + (1-\hat{q})b_0) = (q - \hat{q})(a_0 - b_0) \leq \Delta$$

$$\eta_0 - (\hat{q}a_1 + (1-\hat{q})b_1) = (qa_1 + (1-q)b_1) - (\hat{q}a_1 + (1-\hat{q})b_1) = (q - \hat{q})(a_1 - b_1) \leq \Delta$$

The result of Lemma 2 tells us that

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

Therefore, we must also have $\eta_0, \eta_1 \leq 1/2 - \xi/4 + \Delta$. \hfill \Box

Thus, if we choose the discretization parameter $\Delta$ to be below $\xi/4$, the noise rates will remain bounded away from $1/2$, and the procedure $\text{Lab}$ is a valid example oracle $EX^{n}_{\text{CCCN}}$. If the the concept class $C$ is CN learnable, then we apply the corresponding CCCN learning algorithm to each of the guesses $(\hat{p}, \hat{q})$. The output list of hypotheses is then guaranteed to contain an accurate hypothesis.

**Lemma 4.** Let $\varepsilon, \delta \in (0,1)$. Suppose that the concept class $C$ is CN learnable, and there exists an identified $\xi$-distinguishing event $E$ for the two distributions $P_0$ and $P_1$. Then there exists an algorithm $\mathcal{L}$ such that when given $\varepsilon, \delta, \xi$ 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 $\text{err}(h) \leq \varepsilon$.

**Proof.** Since the concept class $C$ is CN learnable, by the result of [RDM06] we know there exists an efficient algorithm $\mathcal{A}$ that when given access to some example oracle $EX^{n}_{\text{CCCN}}$ with $\eta_0, \eta_1 \leq 1/2 - \xi/8$, outputs a hypothesis $h$ with error bounded $\varepsilon$ with probability at least $1 - \delta$, halts in time $\text{poly}(1/\varepsilon, 1/\delta, 1/\xi, n)$. 

7
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 $i, j \in \{1/\Delta\}$ and $i \neq j$, use the Lab$(\hat{p}, \hat{q}, \xi)$ to generate labeled examples, and run the algorithm $\mathcal{A}$ with sample access to Lab; if the algorithm halts in time $p$ and outputs an hypothesis $\hat{h}$, store the hypothesis in a list $H$. In the end, output the hypothesis list.

By Lemma 3, we know for some guessed values of $p'$ and $q'$, the algorithm 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 $\epsilon$-accurate hypothesis under these guesses. 

Answering an open question in [RDM06] Suppose that the class $\mathcal{C}$ is CN learnable (and therefore CCCN learnable). An open question in [RDM06] asks whether this class is still learnable when the example oracle $EX^c_{CCCN}$ has one noise rate above 1/2 and the other below. If we have no restriction on the noise rates, the answer is trivially no. For example, suppose $\eta_0 = 1 - \eta_1$, then the distributions over the labels are identical for both positive and negative examples and thus the labels are independent of the features $x$. In this case, it is impossible to learn the target concept. However, as long as we can guarantee $|\eta_0 - (1 - \eta_1)| \geq \xi$, then the event of seeing label 0 is a $\xi$-distinguishing event and we can use our algorithm Lab to create a standard CCCN learning instance with both noise rates below 1/2. Therefore, under some mild condition over the noise rates, the answer to this question is yes. We provide details in the appendix.3

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. In order to create a CN learning problem for the target concept $c$, we will assume that the distribution class $\mathcal{P}$ admits a parametric class of distinguishing events.

Assumption 2 (Class of distinguishing events). There exists a parametric class of events $\mathcal{E}(\cdot)$ for the distribution class $\mathcal{P}$ along with a polynomial $p(\cdot, \cdot)$ such that for any $\gamma > 0$ and for any two probability distributions $P$ and $Q$ in $\mathcal{P}$ such that the KL divergence $KL(P \| Q) \geq \gamma$, the class of events $\mathcal{E}(\gamma)$ contains an event $E$ that is an $\xi$-distinguishing for $P$ and $Q$, where $\xi \geq 1/p(k, 1/\gamma)$ and $|\mathcal{E}(\gamma)| \leq p(k, 1/\gamma)$.

Simple Example To give a simple example of such class of events, consider the outcome space $\mathcal{Y} = \{0, 1\}^k$ and the class of full-support product distributions $\mathcal{P}$ over $\mathcal{Y}$. For any two distributions $P, Q \in \mathcal{P}$ such that $KL(P \| Q) \geq \gamma$, under Assumption 1 it can be shown that there exists some coordinate $l$ such that $|P_l - Q_l| \geq 1/poly(k, 1/\gamma)$, 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 feature $j$ is 1 is a candidate distinguishing event, so the class of events is simply $\mathcal{E} = \{1[y_l = 1] | l \in [k]\}$. In the appendix, we also show that a similar construction also works for product distributions over domain $\{0, 1, \ldots, b-1\}^k$ and a special class of multivariate Gaussian distributions.

Given such class of distinguishing events, we will show that the joint class $(\mathcal{C}, \mathcal{P})$ is PwD-learnable.

---

3We only partially resolve this problem because we don't know how to handle unknown noise rates.

4We 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.
Theorem 1. Suppose that the concept class $C$ is CN learnable, and the distribution class $P$ is efficiently learnable and admits a parametric class of events $E$ (Assumption 2). Then the joint class $(C,P)$ is PwD-learnable.

We will present our reduction in three key steps.

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

• Suppose we have learned an accurate hypothesis $h$ from the first step, we can then use $h$ to separate outcomes 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 model $\hat{T} = (h,\hat{P}_0,\hat{P}_1)$.

• 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 $P$) 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.  

4.1 CN Learning with a Class of Events

As a first step in our reduction, we will use the algorithm $L_1$ in Lemma 4: for each event $E$ in the event class $E$, run $L_1$ using $E$ as a candidate distinguishing event. This will again generate a list of possible hypotheses.

Lemma 5. Let $\varepsilon, \delta \in (0,1)$ and $\gamma > 0$. Suppose that the class $C$ is CN learnable, the class $P$ admits a parametric class of events $E$ (as in Assumption 2). If the two distributions $P_0$ and $P_1$ satisfy

$max\{KL(P_0||P_1), KL(P_1||P_0)\} \geq \gamma$, then there exists an algorithm $L_2$ that given access to a class of events $E(\gamma)$, sample access to Gen and $\varepsilon, \delta, \gamma$ as inputs, halts in time bounded by 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 $\text{err}(h) \leq \varepsilon$.

Proof. Consider the following algorithm. We will first use the oracle $E$ with input parameter $\gamma$ to obtain a class of events $E(\gamma)$ that contains a $\xi$-distinguishing event $E^*$ with $\xi \geq \text{poly}(\gamma, 1/n)$. Then for each event $E \in E(\gamma)$, we will run the algorithm $A$ in Lemma 4 with accuracy parameters $\varepsilon, \delta$, separation parameter $\xi$, and $E$ as an hypothetical distinguishing event as input. For each event, the instantiation of algorithm $A$ 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 $\text{err}(h) \leq \varepsilon$ by the guarantee of Lemma 4.  

\[\text{Our result actually extends to any PAC learning algorithm, and we omit the simple details.}\]
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 perturbed. Our result relies on the well-known Neyman-Pearson Lemma in information theory, which is a powerful tool for giving lower bounds in hypothesis testing. We state the following version for our purpose.\(^6\)

**Lemma 6.** [Neyman-Pearson Lemma (see e.g. [Rig15])] Let $Q_0$ and $Q_1$ be two probability distributions over $\mathcal{Y}$, and let $A: \mathcal{Y}^m \rightarrow \{0, 1\}$ be a mapping from $m$ observations in $\mathcal{Y}$ to either 0 or 1. Then

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

where $Y^m \sim Q_0^m$ denotes an i.i.d. sample of size $m$ drawn from the distribution $Q_0$.

In other words, if the two distributions satisfy $\text{KL}(Q_0 \| Q_1) \leq 1/m$, then any procedure that determines whether the observations are drawn from $Q_0$ or $Q_1$ has constant error probability under measure $Q_0$ or $Q_1$. Now let’s construct such a procedure $A$ using a distribution learner $L$ for $\mathcal{P}$. Suppose the learner is $(\varepsilon, \delta)$-accurate given sample of size $m$, then we can define $A$ as the following:

- Run the learning algorithm $L$ 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 $L$.

- If $\text{KL}(Q_0 \| \hat{Q}) \leq \varepsilon$, output 0; otherwise output 1.

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

**Lemma 7.** Let $\varepsilon > 0$, $\delta \in (0, 1/2)$ and $m \in \mathbb{N}$. Suppose there exists a distribution learner $L$ such that for any unknown target distribution $P \in \mathcal{P}$, when $L$ inputs $m$ random draws from $P$, it with probability at least $1 - \delta$ outputs a distribution $\hat{P}$ such that $\text{KL}(P \| \hat{P}) \leq \varepsilon$. Then for any $Q_0 \in \mathcal{P}$ and any distribution $Q_1$ over the same range $\mathcal{Y}$, if the learner $L$ 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 $\text{KL}(Q_0 \| \hat{Q}) \leq \varepsilon$, where $\delta' = \delta + \sqrt{m\text{KL}(Q_0 \| Q_1)/2}$.

**Proof.** To invoke Lemma 6, we will consider the procedure $A$ constructed above that uses the learner $L$ as a subroutine. By the guarantee of the algorithm, we know that $\Pr_{L,Y^m \sim Q_0^m}[\text{KL}(Q_0 \| \hat{Q}) \leq \varepsilon] \geq 1 - \delta$. This means

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

Furthermore, by Lemma 6, we have

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

\(^6\)In the original statement of Neyman-Pearson Lemma, 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.
This in turn implies that with probability at least \((1 - \delta - \sqrt{2\delta KL(Q_0||Q_1)})\) over the random draws of \(Y^m \sim Q_1^m\) and the internal randomness of \(L\), the learner outputs a distribution \(\hat{Q}\) such that \(KL(P||\hat{Q}) \leq \varepsilon\).

Therefore, if the KL divergence between the target distribution and the sample 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 8.** Suppose that the distribution class \(\mathcal{P}\) is efficiently learnable. There exist an algorithm \(L_2\) and a polynomial \(m_P(\cdot,\cdot,\cdot)\) such 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 \(KL(P||Q) \leq 1/(2m_P(1/\varepsilon, 1/\delta, k))\), runs in time \(poly(1/\varepsilon, 1/\delta, k)\) and outputs a list of distributions \(\mathcal{P}'\) that with probability at least \(1 - \delta\) contains some \(\hat{P} \in \mathcal{P}'\) with \(KL(P||\hat{P}) \leq \varepsilon\).

**Proof.** Let \(L\) be a distribution learner that given an independent sample of size \(m\) drawn from the unknown target distribution \(P\), runs in time bounded by \(poly(1/\varepsilon, 1/\delta, n)\) with probability at least \(1 - \delta\), outputs a distribution \(P'\) such that \(KL(P||P') \leq \varepsilon\). By Lemma 7, we know that with probability at least \((1/2 - \delta) \geq 1/4\), the algorithm can also output a distribution \(P''\) such that \(KL(P||P'') \leq \varepsilon\) if the algorithm is given a sample of size \(m\) drawn from the distribution \(Q\).

Let \(r = \log_3(1/\delta)\). 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 \(KL(P||\hat{P}) \leq \varepsilon\).

### 4.3 Learning the Distributions with an Accurate Hypothesis

Now assume that we have obtained an accurate hypothesis \(h\). We will use distribution learner to learn the two distributions \(P_0\) and \(P_1\). For any observation \((x,y)\) drawn from the 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 either distribution. However, because of the classification error of \(h\), the input sample is in fact drawn from a mixture between \(P_0\) and \(P_1\) (with \(err(h)\) weight on the wrong component). 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 algorithm in Lemma 8 to learn each distribution, which gives us a probability model.

First, the following allows us to bound the KL divergence between between a mixture distribution and one of its component.

**Lemma 9.** Let \(P\) and \(Q\) be two distributions over \(\mathcal{X}\) and \(R\) be a mixture of \(P\) and \(Q\) with weights \(w_p\) and \(w_q\) respectively. Then we have \(KL(P||R) \leq w_q KL(P||Q)\).
Proof. Let \( w_p \) and \( w_q \) be the weights associated with \( P \) and \( Q \) respectively in the mixture \( R \).

\[
\text{KL}(P||R) = \int_y P(y) \log \left( \frac{P(y)}{R(y)} \right) dy
\]

\[
= \int_y (w_p P(y) + w_q P(y)) \log \left( \frac{w_p P(y) + w_q P(y)}{w_p P(y) + w_q Q(y)} \right) dy
\]

(by the log-sum inequality)

\[
\leq \int_y (w_p P(y) \log \left( \frac{w_p P(y)}{w_p P(y)} \right)) dy + \int_y (w_q P(y) \log \left( \frac{w_q P(y)}{w_q Q(y)} \right)) dy
\]

\[
= w_q \text{KL}(P||Q)
\]

which proves our claim. \( \square \)

**Lemma 10.** Suppose that the distribution class \( \mathcal{P} \) is efficiently learnable. Let \( \epsilon > 0, 0 < \delta \leq 1 \) 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 \( \epsilon, \delta \) and \( h \) as inputs, \( \mathcal{L}_3 \) runs in time bounded by \( \text{poly}(1/\epsilon, 1/\delta, k) \), and outputs a list of probability models \( T \) such that with probability at least \( 1 - \delta \) there exists some \( \hat{T} \in T \) such that \( \text{err}(\hat{T}) \leq \epsilon \), as long as the hypothesis \( h \) satisfies \( \text{err}(h) \leq 1/r(1/\epsilon, 1/\delta, k) \).

Proof. Our algorithm will first call the oracle \( \text{Gen} \) for \( N = C m_2(2/\epsilon, 4/\delta, k) \left( \frac{M^2}{\epsilon^2} \log(1/\delta) \right) \) 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 8. 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 \( m = m_2(2/\epsilon, 4/\delta) \), the run the learning algorithm \( \mathcal{L}_2 \) in Lemma 8 to the sample with target accuracy \( \epsilon/2 \) 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 \( \pi_j = \Pr_{x \sim D}[h(x) = j] \). Let us first consider the case where \( \pi_j \geq \epsilon/(2M) \). In order to invoke Lemma 9, we will upper bound the quantity \( w_j \text{KL}(P_j||\hat{P}_j) \), where \( w_j = \Pr_{x \sim D}[c(x) = j] \). 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 \( \epsilon_h = \text{err}(h) \), note that when we instantiate the learner \( \mathcal{L}_2 \) on the sample with \( h(x) = j \), the input distribution \( I_j \) is a \((\epsilon_h, 1 - \epsilon_h)\)-mixture of the distributions \( P_{1-j} \) and \( P_j \). Then there exists a polynomial \( r \) such that if \( \text{err}(h) \leq 1/r(1/\epsilon, 1/\delta, k) \), we can have the following based on Lemma 9

\[
\text{KL}(P_j||I_j) \leq \epsilon_h \text{KL}(P||Q) \leq 1/mr(2/\epsilon, 4/\delta, k)
\]

where \( mr \) is the polynomial defined in Lemma 8. 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 \( \text{KL}(P_j||\hat{P}_j) \leq \epsilon/2 \), which implies that \( w_j \text{KL}(P_j||\hat{P}_j) \leq \epsilon/2 \).

Suppose that \( \pi_j < \epsilon/(2M) \), then we know that no matter what the distribution \( \hat{P}_j \) is, we have \( w_j \text{KL}(P_j||\hat{P}_j) \leq \frac{\epsilon/2}{2M} = \epsilon/2 \) by Assumption 1.

Finally, our algorithm will output a list of probability models \( T = \{(h_0, \hat{P}_0, \hat{P}_1) | \hat{P}_0 \in \mathcal{P}_0, \hat{P}_1 \in \mathcal{P}_1 \} \), such that with probability at least \( 1 - \delta \), there exists some model \( \hat{T} = (h_0, \hat{P}_0, \hat{P}_1) \in T \) such that

\[
\text{err}(T) = w_0 \text{KL}(P_0||\hat{P}_0) + w_1 \text{KL}(P_1||\hat{P}_1) \leq \epsilon,
\]

which recovers our claim. \( \square \)
4.4 Directly Applying the Distribution Learner

Lastly, we show that if the two distributions \( P_0 \) and \( P_1 \) are too close to admit a distinguishing event, we do not actually need to learn the target concept \( c \) to obtain an accurate probability model — we can simply run the distribution learner in Lemma \( 8 \) over the samples drawn from the mixture to learn single distribution. Our result also extends to the setting where the weight of one component is close to 0, which will be useful for the reverse reduction in Section 5.

**Lemma 11.** Suppose that the distribution class \( \mathcal{P} \) is efficiently learnable. Let \( R \) be the mixture distribution over the outcomes \( y \)'s under the distribution \( \text{Gen} \) (with weights \( w_0 \) and \( w_1 \)). 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 \( \text{Gen} \) and \( \varepsilon,\delta \) as inputs, it runs in time bounded by \( \text{poly}(1/\varepsilon,1/\delta,k) \) and as long as \( \min\{w_0, w_1\} \leq 1/g(1/\varepsilon,1/\delta,k) \) or \( \max\{\text{KL}(P_0||P_1),\text{KL}(P_1||P_0)\} \leq 1/g(1/\varepsilon,1/\delta,k) \) holds, it will with probability at least \( 1 - \delta \), output a list of distributions \( \mathcal{P}' \) that contains \( \hat{P} \) with \( \mathbb{E}_{x \sim \mathcal{D}}[\text{KL}(P_\gamma||\hat{P})] \leq \varepsilon \).

**Proof.** We first consider the case where the weight on one component is small, and without loss of generality assume that \( w_1 \leq \varepsilon/(4Mm) \). By Lemma 9 and Assumption 1, we know that

\[
\text{KL}(P_0||R) \leq w_1 \text{KL}(P_0||P_1) \leq \frac{\varepsilon}{2Mm} M \leq 1/(2m).
\]

By instantiating the algorithm in Lemma \( 8 \) with parameters \( (\varepsilon/2, \delta) \), we know with probability \( 1 - \delta \), there exists a hypothesis distribution \( \hat{P} \) in the output list such that \( \text{KL}(P_0||\hat{P}) \leq \varepsilon/2 \). Again by our Assumption 1, we know \( \text{KL}(P_1||\hat{P}) \leq M \), so it follows that

\[
\mathbb{E}_{x \sim \mathcal{D}}[\text{KL}(P_\gamma||\hat{P})] = w_0 \text{KL}(P_0||\hat{P}) + w_1 \text{KL}(P_1||\hat{P}) \leq \frac{\varepsilon}{2} + \frac{\varepsilon M}{2Mm} \leq \varepsilon.
\]

Next suppose that we are in the second case where \( \text{KL}(P_0||P_1),\text{KL}(P_1||P_0) \leq 1/(2m) \). We know from Lemma 9 that

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

We will also apply the algorithm in Lemma \( 8 \) which guarantees with probability at least \( 1 - \delta \) that there exists a hypothesis distribution \( \hat{P} \) in the output list \( \mathcal{P}' \) such that \( \text{KL}(P_0||\hat{P}),\text{KL}(P_1||\hat{P}) \leq \varepsilon/2 \), which implies that

\[
\mathbb{E}_{x \sim \mathcal{D}}[\text{KL}(P_\gamma||\hat{P})] = w_0 \text{KL}(P_0||\hat{P}) + w_1 \text{KL}(P_1||\hat{P}) \leq \varepsilon.
\]

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 \( \varepsilon,\delta \) and \( m \).

**Proof of Theorem 1** We will now combine all the tools to prove Theorem 1. First, consider the class of events \( \mathcal{E}(\gamma) \) with \( \gamma = 1/g(1/\varepsilon,1/\delta,k) \) (specified in Lemma 11). Then we will apply the CN algorithm \( \mathcal{L}_2 \) in Lemma 5 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 \( T \). If \( \max\{\text{KL}(P_0||P_1),\text{KL}(P_1||P_0)\} \geq \gamma \), then \( T \) is guaranteed to contain an \( \varepsilon \)-accurate model with high probability (based on Lemmas 5 and 10). Next, apply the distribution learner in Lemma 11 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 \( \max(\text{KL}(P_0 || P_1), \text{KL}(P_1 || P_0)) < \gamma \), we know \( T' \) is \( \varepsilon \)-accurate with high probability (based on Lemma 11). Finally, apply the maximum likelihood method to the list of models \( T \cup \{T'\} \): draw a sample of polynomial size from Gen, then for each model \( T \in T \cup \{T'\} \), 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. (See the appendix for details.)

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 1 can be applied. There are other important cases — such as general multivariate Gaussians — for which we do not know such classes.\(^7\) 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 [FSO06, Das99].

## 5 Reverse Reduction

In our reverse\(^8\) reduction, our strategy is to first learn the two distributions \( P_0 \) and \( P_1 \) sufficiently well, and then use a CCCN learner for learning the target concept \( c \). We will make a stronger learnability assumption on the distribution class \( \mathcal{P} \) — we assume that we have a parametrically correct learner for any mixture of two distributions in \( \mathcal{P} \).

**Assumption 3** (Parametrically correct mixture learning). There exists an algorithm \( \mathcal{L}_M \) and a polynomials \( \rho \) such that for any \( Z \) that is an unknown \( \pi \)-mixture of two distributions \( Y_0 \) and \( Y_1 \) from the class \( \mathcal{P} \), and for any \( \varepsilon > 0, 0 < \delta \leq 1 \) 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 \( \text{poly}(k, 1/\varepsilon, 1/\delta) \) and outputs a mixture \( \hat{Z} \) of distributions \( \hat{Y}_0 \) and \( \hat{Y}_1 \) such that \( \text{KL}(Z \| \hat{Z}') \leq \varepsilon \). Furthermore, if the “healthy mixture” condition holds: \( \min(\pi_0, \pi_1) \geq 1/\rho(1/\varepsilon, 1/\delta, k) \) and \( \max(\text{KL}(Y_0 \| Y_1), \text{KL}(Y_1 \| Y_0)) \geq 1/\rho(1/\varepsilon, 1/\delta, k) \), the output distributions \( \hat{Y}_0 \) and \( \hat{Y}_1 \) satisfy \( \text{KL}(Y_0 \| \hat{Y}_0) \leq \varepsilon \) and \( \text{KL}(Y_1 \| \hat{Y}_1) \leq \varepsilon \).

We remark that the assumption of parametric correctness is a mild condition, and is satisfied by almost all mixture learning algorithms in the literature. Also note that we only require this condition when the healthy mixture condition is met. If 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 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 3). Then the joint class \( (\mathcal{C}, \mathcal{P}) \) is PwD-learnable.

Since we can use the mixture learner to learn the two distributions \( P_0 \) and \( P_1 \), the only obstacle for obtaining a good probability model is to learn the target concept \( c \). Our approach is to use the accurate distributions \( \hat{P}_0 \) and \( \hat{P}_1 \) output by the mixture learner to create a specialized distinguishing event, which allows us to learn \( c \) with a CN learner. In the case where we cannot

---

\(^7\)We conjecture that Gaussians do indeed have a small set of distinguishing events, but have not been able to prove it.

\(^8\)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.
learn component accurately because the mixture is unbalanced or the two distributions are close, we can again appeal to the robustness result we show using Neyman-Pearson lemma — 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 $\tau$, let 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 two distributions are different.

**Lemma 12.** Fix any $\gamma \in (0, 1]$. Suppose that $KL(P||Q) \geq \gamma$, 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 1, and for any $y \notin E$, we also have $\log \frac{P(y)}{Q(y)} < \gamma/2$.

\[
KL(P||Q) = \int_{y \in \mathcal{Y}} P(y) \log \frac{P(y)}{Q(y)} dy
\]
\[
= \int_{y \in E} P(y) \log \frac{P(y)}{Q(y)} dy + \int_{y \notin E} P(y) \log \frac{P(y)}{Q(y)} dy
\]
\[
< P(E)M + (1 - P(E))\frac{\gamma}{2}
\]
\[
= \frac{\gamma}{2} + (M - \gamma/2)P(E) < \frac{\gamma}{2} + M P(E)
\]
Since we know that $\text{KL}(P\|Q) \geq \gamma$, it follows that $P(E) > \frac{\gamma}{2M}$. Furthermore,

\[
P(E) - Q(E) = P(E) \left(1 - \frac{Q(E)}{P(E)}\right) \\
\geq P(E) \left(1 - \sup_{y \in E} \frac{Q(y)}{P(y)}\right) \\
\geq P(E) \left(1 - 2^{-\gamma/2}\right) \geq \frac{\gamma P(E)}{4}
\]

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

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

which proves our statement. \hfill \Box

Next, we will 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 enough.

**Lemma 13.** Suppose that the distributions $P, \hat{P}, Q, \hat{Q}$ over $\mathcal{Y}$ satisfy that $\text{KL}(P\|\hat{P}) \leq \alpha$, $\text{KL}(Q\|\hat{Q}) \leq \alpha$, and $\text{KL}(P\|Q) \geq \gamma$ for some $\alpha, \gamma \in (0,1]$. Then the event $E(\hat{P}, \hat{Q}, (\gamma^2/(8M) - \sqrt{2\alpha})^2)$ is a $\xi$-distinguishing event with $\xi \geq 1/\text{poly}(1/\gamma, 1/\alpha, k)$ as long as $\gamma > 8M(\sqrt{2\alpha} + (8M^2 \alpha)^{1/8})$.

**Proof.** Since we have both $\text{KL}(P\|\hat{P}), \text{KL}(Q\|\hat{Q}) \leq \alpha$, 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 12 and the definition of total variation distance, we know that

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

By triangle inequality, the above implies

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

By Pinsker’s inequality, we know that $\|\hat{P} - \hat{Q}\|_{tv} \leq \sqrt{\text{KL}(\hat{P}\|\hat{Q})/2}$. It follows that $\text{KL}(\hat{P}\|\hat{Q}) \geq 2b^2$. Consider the event $E = E(\hat{P}, \hat{Q}, b^2)$. We know by Lemma 12 that $E$ is a $(b^4/(2M))$-distinguishing event for distributions $\hat{P}$ and $\hat{Q}$. Since both $\text{KL}(P\|\hat{P}), \text{KL}(Q\|\hat{Q}) \leq \alpha$, we have

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

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

\[
|P(E) - Q(E)| = |(P(E) - \hat{P}(E)) + (\hat{P}(E) - \hat{Q}(E)) + (\hat{Q}(E) - Q(E))| \\
\geq |\hat{P}(E) - \hat{Q}(E)| - |P(E) - \hat{P}(E)| - |\hat{Q}(E) - Q(E)| \\
\geq (b^4/(2M)) - \sqrt{2\alpha}.
\]

Note that if we have $\gamma > 8M(\sqrt{2\alpha} + (8M^2 \alpha)^{1/8})$, then we can guarantee both $b > 0$ and $(b^4/(2M)) - \sqrt{2\alpha} > 0$. \hfill \Box
Given these structural lemmas, we will focus on the case where the mixture learner can return accurate approximations — that is, the marginal distribution over \( \mathcal{Y} \) is a healthy mixture of two sufficiently different distributions. Specifically, we can use the mixture learner to approximate \( P_0 \) and \( P_1 \) well enough, so that we can use the approximations to create a distinguishing event. Once we have such an event, we can then create a CN learning problem and use the algorithm in Lemma 4 to learn the concept \( c \).

**Lemma 14.** Suppose the class \( \mathcal{P} \) satisfies the parametric mixture learning assumption (Assumption 3), the class \( \mathcal{C} \) is CN learnable, and there exists some \( \gamma > 0 \) such that

\[
\min\{w_0, w_1\} \geq \gamma \quad \text{and} \quad \max\{\text{KL}(P_0||P_1), \text{KL}(P_1||P_0)\} \geq \gamma.
\]

Then there exists an algorithm \( L \) that given \( \epsilon, \delta \) and \( \gamma \) as inputs and sample access from Gen, halts in time bounded by \( \text{poly}(1/\epsilon, 1/\delta, 1/\gamma, n, k) \), and with probability at least \( 1 - \delta \), outputs a list of probability models \( T \) that contains some \( \hat{T} \) with

\[
\text{err}(\hat{T}) \leq \epsilon.
\]

**Proof.** We will first invoke the algorithm \( \mathcal{L}_M \) in Assumption 3 so that with probability at least \( 1 - \delta/2 \), the output approximations for the two components satisfy \( \text{KL}(P_0||\hat{P}_0) \leq \alpha \) and \( \text{KL}(P_1||\hat{P}_1) \leq \alpha \) for some \( \alpha \) that satisfies \( \gamma > 8M(\sqrt{2\alpha} + (8M^2 \alpha)^{1/8}) \). This process will halt in time \( \text{poly}(1/\alpha, 1/\delta, 1/\gamma, k) \).

By Lemma 12, we know that the either event \( E(\hat{P}_0, \hat{P}_1, \gamma/2) \) is a \( \xi \)-distinguishing event for \( P_0 \) and \( P_1 \) for some \( \xi \geq 1/\text{poly}(1/\gamma, n, k) \). Then we can use the CN learning algorithm \( \mathcal{L}_1 \) in Lemma 4 with the distinguishing event \( E \) to learn a list of hypotheses \( H \) under polynomial time, and there exists some \( h \in H \) that is \( \epsilon_1 \) accurate, with \( \epsilon_1 = 1/r(1/\epsilon, 1/\delta, k) \) (specified in Lemma 10). For each hypothesis \( h' \in H \), run the algorithm \( \mathcal{L}_3 \) with \( h' \) as the candidate hypothesis and \( \epsilon \) as the target accuracy parameter. By Lemma 10, this will halt in polynomial time, and outputs a list of probability models \( T \) such that one of which has error \( \text{err}(\hat{T}) \leq \epsilon \).

Finally, to wrap up and prove Theorem 2, we also need to handle the case where the mixture learner fails to approximate both components. We will again appeal to the robust distribution learner in Lemma 11 to learn the distributions directly. In particular, if the healthy mixture condition does not hold, we can simply predict with the distribution output by the robust learner.

**Proof of Theorem 2** The algorithm consists of three steps. First, we will run the algorithm in Lemma 14 by setting \( \gamma = 1/g(1/\epsilon, \delta, k) \) (specified in Lemma 10) and other parameters in a way to guarantee that whenever \( \max\{\text{KL}(P_0||\hat{P}_0), \text{KL}(P_1||\hat{P}_1)\} \geq \gamma \) and \( \min\{w_0, w_1\} \geq \gamma \) both hold, the output list of models \( T \) contains some \( T \) that has error at most \( \epsilon \). Next, we will directly apply the distribution learner in Lemma 11 so that when the healthy mixture condition is not met, the algorithm outputs a distribution \( \hat{P} \) such that \( \mathbb{E}_{x \sim \mathcal{D}} [\text{KL}(P_{c(x)}||\hat{P})] \). Lastly, similar to the final step in the forward reduction, we run the maximum likelihood algorithm to output the model in \( T \cup \{h_{0}, \hat{P}, \hat{P}\} \) with the smallest empirical log-loss. (See appendix for details.)

### 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.

**Acknowledgements**  We thank Akshay Krishnamurthy and Shahin Jabbari for helpful discussions.
References

[AL87] Dana Angluin and Philip D. Laird. Learning from noisy examples. *Machine Learning*, 2(4):343–370, 1987.

[Das99] 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.

[Dec97] 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.

[FOS08] 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.

[FSO06] 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.

[Hau92] David Haussler. Decision theoretic generalizations of the PAC model for neural net and other learning applications. *Inf. Comput.*, 100(1):78–150, 1992.

[Kea98] Michael J. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.

[KMR+94] 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, 1994.

[RDM06] 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.

[Rig15] Philippe Rigollet. High dimensional statistics, 2015.

[Val84] 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.
A 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 $\mathbb{E}_{x \sim D} \left[ KL(P_{c(x)}||\hat{P}_{h(x)}) \right]$ 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 $\text{loss}(y, \hat{P}) = -\log \hat{P}(y)$. 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

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

(3)

which will be useful for proving the next lemma.

Lemma 15. Given any hypothesis $h: \mathcal{X} \rightarrow \{0, 1\}$, and hypothesis distributions $\hat{P}_0$ and $\hat{P}_1$, we have

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

Proof. We can write the following

$$\mathbb{E}_{x \sim D} \left[ KL(P_{c(x)}||\hat{P}_{h(x)}) \right] = \mathbb{E}_{D} \left[ \Pr[c(x) = 1, h(x) = 1] KL(P_1||\hat{P}_1) + \Pr[c(x) = 1, h(x) = 0] KL(P_1||\hat{P}_0) \right]$$

$$+ \mathbb{E}_{D} \left[ \Pr[c(x) = 0, h(x) = 1] KL(P_0||\hat{P}_1) + \Pr[c(x) = 0, h(x) = 0] KL(P_0||\hat{P}_0) \right]$$

(apply Equation (3))

$$= \mathbb{E}_{x \sim D} \left[ H(P_{c(x)}) \right] - \sum_{(i,j) \in \{0, 1\}^2} \mathbb{E}_{D} \left[ \Pr[c(x) = i, h(x) = j] \mathbb{E}_{y \sim \hat{P}_j} \left[ \log(\hat{P}_j(y)) \right] \right]$$

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

which proves our claim. \qed

Therefore, we could write $\text{err}(T) = \mathbb{E}_{x \sim D} \left[ H(P_{c(x)}) \right] - \mathbb{E}_{(x,y) \sim \text{Gen}} \left[ \log(\hat{P}_{h(x)}(y)) \right]$ for any model $T = (h, \hat{P}_0, \hat{P}_1)$. Observe that $\mathbb{E}_{x \sim D} \left[ H(P_{c(x)}) \right]$ 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 $T$ as input, draw a set of $S$ of samples $(x,y)$’s from Gen, and for each $T = (h, \hat{P}_0, \hat{P}_1) \in T$, compute the log-loss on the sample

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

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

Our goal is to show that if the list of models $T$ contains an accurate model $T$, the maximum likelihood algorithm will then output an accurate model with high probability.
**Theorem 3.** Let $\epsilon > 0$. Let $T$ be a set of probability models such that at least one model $T^* \in T$ has error $\text{err}(T^*) \leq \epsilon$. Suppose that the class $\mathcal{P}$ also satisfies bounded assumption (in Assumption 1).

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

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

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

**Theorem 4.** 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

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

**Proof.** Our proof essentially follows from the same analysis of [FOS08] (Theorem 17). We say that a probability model $T$ is good if $\text{err}(T) \leq 4\epsilon$, and bad otherwise. We know that $T$ is guaranteed to contain at least one good model. In the following, we will write $H(\text{Gen})$ to denote $\mathbb{E}_{X \sim D}\left[H(P_{\text{Gen}}(X))\right]$.

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

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

For each bad model $T'$ with $\text{err}(T') > 4\epsilon$, we can write

$$
\Pr[\text{loss}(S, T') \leq m(H(\text{Gen}) + 3\epsilon)] = \Pr[\text{loss}(S, T') \leq m(H(\text{Gen}) + 4\epsilon) - \epsilon m]
$$

(because $\text{err}(T') \geq 0$)

$$
\leq \Pr[\text{loss}(S, T') \leq m(H(\text{Gen}) + \text{err}(T') - \epsilon m)]
$$

$$
= \Pr[\text{loss}(S, T') \leq \mathbb{E}_{\text{Gen}^m}\left[\text{loss}(S, T') - \epsilon\right]]
$$

$$
\leq \exp\left(-\frac{2m\epsilon^2}{M^2}\right)
$$

where the last step follows from Theorem 4. Similarly, for the best model $T^*$ with $\text{err}(T^*) \leq \epsilon$, we have the following derivation:

$$
\Pr[\text{loss}(S, T^*) \geq m(H(\text{Gen}) + 2\epsilon)] = \Pr[\text{loss}(S, T^*) \geq m(H(\text{Gen}) + \epsilon) + m\epsilon]
$$

$$
\leq \Pr[\text{loss}(S, T^*) \geq m(H(\text{Gen}) + \text{err}(T^*) + m\epsilon)]
$$

$$
= \Pr[\text{loss}(S, T^*) \geq \mathbb{E}_{\text{Gen}^m}\left[\text{loss}(S, T^*)]\right] + m\epsilon
$$

$$
\leq \exp\left(-\frac{2m\epsilon^2}{M^2}\right)
$$

Combining these two probabilities recovers the stated bound. $\Box$

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

21
B Solution to an Open Problem in [RDM06]

Now we consider a question left open by [RDM06]. Suppose the CCCN example oracle $EX^\eta_{CCCN}$ has noise rates $\eta_0$ and $\eta_1$ such that one of them might be above $1/2$ and the other might be below $1/2$, can we still learn the concept class $C$ assuming that it is CN learnable? We will resolve this problem in two ways.

B.1 Impossibility Results

If we make no restriction on the problem instance, the answer is no — it is information theoretically impossible to learn the target concept for the following two reasons:

- If the noise rates happen to satisfy $\eta_0 = 1 - \eta_1$, then examples drawn from the oracle $EX^\eta_{CCCN}$
  $$\Pr[y = 1 | c(x) = 1] = \Pr[y = 1 | c(x) = 0] = \Pr[y = 1] = \eta_0.$$  
  This means the the label $y$ is independent of the feature $x$, and conveys no information about the target concept $c$. Therefore, it is impossible distinguish the target concept $c$ from any other concept in the class.

- If the class $C$ is close under complements (for any $c \in C$, $\neg c$ is also in $C$), then it is in general impossible to differentiate a concept $c$ from its negation $\neg c$. For any concept $c$ and values $p, q \in [0, 1]$, we can create the following two learning instances:
  1. The target concept is $c$, and the noise rates are $\eta_0 = p$ and $\eta_1 = 1 - q$.
  2. The target concept is $\neg c$, and the noise rates are $\eta_0 = q$ and $\eta_1 = 1 - p$.

  It is easy to check in these two instances, their example oracles will induce identical joint distribution over the labeled examples $(x, y)$.

Therefore, even if we are allowed to have unbounded computation time, there are learning instances in which it is impossible to learn an accurate hypothesis.

B.2 Upper Bound

Given the impossibility result above, we know that in order to learn under this general noise setting, it is necessary to assume that $\eta_0$ and $(1 - \eta_1)$ are sufficiently different. Even if we make such an assumption, we need more information to identify which part of the data is positive, otherwise we cannot distinguish the target concept from its negation even when we have learned a arbitrarily accurate partition. Therefore, we assume we know the values of $\eta_1$ and $\eta_0$ (or some $\varepsilon$-accurate estimates of them with $\varepsilon < \xi/4$).

Our result follows from directly applying the Lemma 3 — there exists an polynomial-time algorithm that learns an accurate hypothesis as long as the class $C$ is CN learnable and the noise rates satisfy $|\eta_1 - (1 - \eta_0)| \geq \xi$ for some value $\xi > 0$.

**Theorem 5.** Let $\varepsilon, \delta \in (0, 1/2)$. Suppose that the class $C$ is CN learnable. Let $c$ be any unknown target concept in $C$ and $EX^\eta_{CCCN}$ be an example oracle with noise rates $\eta_0$ and $\eta_1$ such that $|\eta_1 - (1 - \eta_0)| \geq \xi$ for some value $\xi \in (0, 1]$. Let estimates for the noise rates $\hat{\eta}_0$ and $\hat{\eta}_1$ be such that $\Delta = \max\{|\hat{\eta}_0 - \eta_0|, |\hat{\eta}_1 - \eta_1|\} < \xi/4$. Then there exists an algorithm $L$ that given sample access to $EX^\eta_{CCCN}$ and $\varepsilon, \delta, \xi, \hat{\eta}_0, \hat{\eta}_1$ as inputs,
halts in time \( poly(1/\varepsilon, 1/\delta, 1/(\xi/4 - \Delta), n) \) and with probability at least \( 1 - \delta \) outputs a hypothesis \( h \in C \) such that \( err(h) \leq \varepsilon \).

Proof. We can instantiate the algorithm \( \text{Lab}(\hat{p}, \hat{q}, \xi) \) with \( \hat{p} = \eta_0, \hat{q} = 1 - \eta_1 \) and \( y = 1 \) as a \( \xi \)-distinguishing event. By Lemma 3, this will give us an oracle \( EX_{\text{CCCN}}^{\eta} \) with noise rates below 1/2, which allows us to invoke a CCCN learner to learn the target concept \( c \). \( \square \)

C Examples of Distinguishing Events

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

C.1 Spherical Gaussians

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

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

where \( I \) denotes the identity matrix in \( \mathbb{R}^k \).

**Theorem 6.** There exists a parametric class of events \( E(\cdot) \) for the distribution class \( P \) of \( k \)-dimensional Spherical Gaussians such that for any \( \gamma > 0 \) and for any two probability distributions \( P \) and \( Q \) in the class \( P \) such that \( KL(P\|Q) \geq \gamma \), the class of events \( E(\gamma) \) contains an event \( E \) that is a \( \xi \)-distinguishing event, where

\[
\max\{1/\xi, |E(\gamma)|\} \leq poly(k,1/\gamma).
\]

**Proof.** Recall that the KL divergence of two multivariate Gaussian distributions \( P \) and \( Q \) with means \( \mu, \mu' \) and covariance matrices \( \Sigma_p, \Sigma_q \) can be written as

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

For any two distributions \( P \) and \( Q \) in our class \( P \), we can simplify the KL divergence as

\[
KL(P\|Q) = \frac{1}{2} ||\mu - \mu'||_2^2.
\]

Then \( KL(P\|Q) \geq \gamma \) implies that there exists some coordinate \( j \in [k] \) such that \( |\mu_j - \mu'_j| \geq \sqrt{2\gamma/k} \). Note that the marginal distributions of \( P_j \) and \( Q_j \) over the \( j \)-the coordinate are \( \mathcal{N}(\mu_j,1) \) and \( \mathcal{N}(\mu'_j,1) \) respectively. Without loss of generality, assume that \( \mu'_j < \mu_j \). Then for any value \( t \in [\mu'_j, \mu_j] \), we have

\[
P_j[y \geq t] - Q_j[y \geq t] \geq P_j[y \in [t, \mu_j]].
\]

Let \( \Delta = \sqrt{2\gamma/k} \), and consider the discretized set \( L(\gamma) = \{0, \Delta, \ldots, [1/\Delta]\Delta\} \). 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 (4), we can write

\[
P_j[y \geq t'] - Q_j[y \geq t'] \geq \frac{1}{2} \text{erf}(\Delta)
\]
where \( \text{erf} \) denotes the Gauss error function with \( \text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-a^2} \, da \) for every \( x \in \mathbb{R} \). The Taylor expansion of the function is

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

Therefore, for any \( x \in [0,1) \), there exists a constant \( C \) such that \( \text{erf}(x)/2 \geq C x \). 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

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

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

\[\square\]

### 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 \( M \leq \text{poly}(k,b) \) 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 7.** 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 \( \text{KL}(P||Q) \geq \gamma \), the class of events \( \mathcal{E}(\gamma) \) contains an event \( E \) that is an \( \xi \)-distinguishing event, where \( \max\{1/\xi,|\mathcal{E}(\gamma)|\} \leq \text{poly}(k,b,1/\gamma) \).

**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

\[
\text{KL}(P||Q) = \sum_{j \in [k]} \text{KL}(P_j||Q_j).
\]

Therefore \( \text{KL}(P||Q) \geq \gamma \) implies that there exists some coordinate \( j \) such that \( \text{KL}(P_j||Q_j) \geq \gamma/k \). 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 \( \log(P_j(t)/Q_j(t)) \leq M \), 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

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

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

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

would contain a \( \xi \)-distinguishing event, and \( \max\{1/\xi,|\mathcal{E}(\gamma)|\} \leq \text{poly}(k,b,1/\gamma) \). \[\square\]