Proceedings of Machine Learning Research vol 75:1–19, 2018 31st Annual Conference on Learning Theory

Actively Avoiding Nonsense in Generative Models

Steve Hanneke STEVE.HANNEKE@GMAIL.COM
Princeton, NJ

Adam Tauman Kalai ADUM@MICROSOFT.COM
Microsoft Research, New England

Gautam Kamath G@CSAIL.MIT.EDU
EECS & CSAIL, MIT

Christos Tzamos CHTZAMOS@MICROSOFT.COM
Microsoft Research, New England

Editors: Sebastien Bubeck, Vianney Perchet and Philippe Rigollet

Abstract

A generative model may generate utter nonsense when it is fit to maximize the likelihood of observed data. This happens due to “model error,” i.e., when the true data generating distribution does not fit within the class of generative models being learned. To address this, we propose a model of active distribution learning using a binary invalidity oracle that identifies some examples as clearly invalid, together with random positive examples sampled from the true distribution. The goal is to maximize the likelihood of the positive examples subject to the constraint of (almost) never generating examples labeled invalid by the oracle. Guarantees are agnostic compared to a class of probability distributions. We first show that proper learning may require exponentially many queries to the invalidity oracle. We then give an improper distribution learning algorithm that uses only polynomially many queries.

Keywords: Generative models, active learning, statistical learning

1. Introduction

Generative models are often trained in an unsupervised fashion, fitting a model q to a set of observed data $x_P \subseteq X$ drawn iid from some true distribution p on $x \in X$. Now, of course p may not exactly belong to family Q of probability distributions being fit, whether Q consists of Gaussians mixture models, Markov models, or even neural networks of bounded size. We first discuss the limitations of generative modeling without feedback, and then discuss our model and results.

Consider fitting a generative model on a text corpus consisting partly of poetry written by four-year-olds and partly of mathematical publications from the *Annals of Mathematics*. Suppose that learning to generate a poem that looks like it was written by a child was easier than learning to generate a novel mathematical article with a correct, nontrivial statement. If the generative model pays a high price for generating unrealistic examples, then it may be better off learning to generate children’s poetry than mathematical publications. However, without negative feedback, it may be difficult for a neural network or any other model to know that the mathematical articles it is generating are stylistically similar to the mathematical publications but do not contain valid proofs.[1]

[1] This is excluding clearly fake articles published without proper review in lower-tier venues (Labbé and Labbé, 2013).

As a simpler example, the classic Markovian "trigram model" of natural language assigns each word a fixed probability conditioned only on the previous two words. Prior to recent advances in deep learning, for decades the trigram model and its variant were the workhorses of language modeling, assigning much greater likelihood to natural language corpora than numerous linguistically motivated grammars and other attempts (Rosenfeld, 2000). However, text sampled from a trigram is typically nonsensical, e.g., the following text was randomly generated from a trigram model fit on a corpus of text from the Wall Street Journal (Jurafsky and Martin, 2009):

> They also point to ninety nine point six billion dollars from two hundred four oh six three percent of the rates of interest stores as Mexico and gram Brazil on market conditions.

In some applications, like text compression using a language model (Witten et al., 1987), maximizing likelihood is equivalent to optimizing compression. However, in many applications involving generation, such nonsense is costly and unacceptable. Now, of course it is possible to always generate valid data by returning random training examples, but this is simply overfitting and not learning. Alternatively, one could incorporate human-in-the-loop feedback such as through crowdsourcing, into the generative model to determine what is a valid, plausible sentence.

In some domains, validity could be determined automatically. Consider a Markovian model of a well-defined concept such as mathematical formulas that compile in LaTeX. Now, consider a n-gram Markovian character model which the probability of each subsequent character is determined by the previous n characters. For instance, the expression ${2+{x-y}$ is invalid in LaTeX due to mismatched braces. For this problem, a LaTeX compiler may serve as a validity oracle. Various n-gram models can be fit which only generate valid formulas. To address mismatched braces, for example, one such model would ensure that it always closed braces within n characters of opening, and had no nested braces. While an n-gram model will not perfectly model the true distribution over valid LaTeX formulas, for certain generative purposes one may prefer an n-gram model that generates valid formulas over one that assigns greater likelihood to the training data but generates invalid formulas.

Figure 1 illustrates a simple case of learning a rectangle model for data which is not uniform over a rectangle. A maximum likelihood model would necessarily be the smallest rectangle containing all the data, but most examples generated from this distribution may be invalid. Instead a smaller rectangle, as illustrated in the figure, may be desired.

Motivated by these observations, we evaluate a generative model q on two axes. First is *coverage*, which is related to the probability assigned to future examples drawn from the true distribution p. Second is *validity*, defined as the probability that random examples generated from q meet some validity requirement. Formally, we measure coverage in terms of a bounded *loss*:

$$\mathrm{Loss}(p, q) = \mathbf{E}_{x \sim p}[L(q_x)],$$

where $L : [0, 1] \to [0, M]$ is a bounded decreasing function such as the capped log-loss $L(q_x) = \min(M, \log 1/q_x)$. A bounded loss has the advantages of being efficiently estimable, and also it enables a model to assign 0 probability to one example (e.g., an outlier or error) if it greatly increases the likelihood of all other data. Validity is defined with respect to a set $V \subseteq X$, and $q(V)$ is the probability that a random example generated from q lies within V.

Clearly, there is a tradeoff between coverage and validity. We first focus on the case of (near) perfect validity. A Valid Generative Modeling (VGM) algorithm if it outputs, for a family of distributions Q over X, if it outputs $\hat{q}$ with (nearly) perfect validity and whose loss is nearly as good as

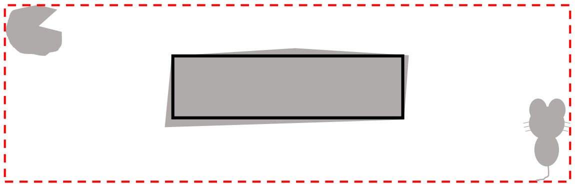

Figure 1: Example where the underlying distribution p is uniform over the (gray) valid regions. The solid rectangle maximizes our objective since it does not output nonsense (is supported only within the grey matter) and is closest to the p (covers the maximum amount of grey matter). In contrast, the standard maximum likelihood (dashed red) rectangle must fully contain the observed samples, thus generating invalid points most of the time.

the loss of the best valid $q \in Q$. More precisely, A is a VGM learner of Q if for any nonempty valid subset $V \subseteq X$, any probability distribution p over V, and any $\varepsilon > 0$, A uses n random samples from p and makes m membership oracle calls to V and outputs a distribution $\hat{q}$ such that,

$$\text{Loss}(p, \hat{q}) \leq \min_{q \in Q : q(V)=1} \text{Loss}(p, q) + \varepsilon \;\text{ and }\; \hat{q}(V) \geq 1 - \varepsilon.$$

We aim for our learner to be sample and query efficient, requiring that n and m are polynomial in $M, 1/\varepsilon$ and a measure of complexity of our distribution class Q. Furthermore, we would like our algorithms to be computationally efficient, with a runtime polynomial in the size of the data, namely the $n + m$ training examples. A more formal description of the problem is available in Section 2.

A is said to be *proper* if it always outputs $\hat{q} \in Q$ and *improper* otherwise. In Section 3.2, we first show that efficient proper learning for VGM is impossible. This is an information-theoretic result, meaning that even given infinite runtime and positive samples, one still cannot solve the VGM problem. Interestingly, this is different from binary classification, where it is possible to statistically learn from iid examples without a membership oracle.

Our first main positive result is an efficient (improper) learner for VGM. The algorithm relies on a subroutine that solves the following *Generative Modeling with Negatives* (GMN) problem: given sets $X_P, X_N \subset X$ of positive and negative examples, find the probability distribution $q \in Q$ which minimizes $\sum_{x \in X_P} L(q(x))$ subject to the constraint that $q(X_N) = 0$. For simplicity, we present our algorithm for the case that the distribution family Q is finite, giving sample and query complexity bounds that are logarithmic in terms of $|Q|$. However, as we show in Section 5.3, all of our results extend to infinite families Q. It follows that if one has a computationally efficient algorithm for the GMN problem for a distribution family Q, then our reduction gives a computationally efficient VGM learning algorithm for Q.

Our second positive result is an algorithm that minimizes $\text{Loss}(p, q)$ subject to a relaxed validity constraint comparing against the optimal distribution that has validity $q(V)$ at least $1 - \alpha$ for some $\alpha > 0$. We show in Section 5.1 that even in this more general setting, it is possible to obtain an algorithm that is statistically efficient but may not be computationally efficient. An important open question is whether there exists a computationally efficient algorithm for this problem when given access to an optimization oracle, as was the case for our algorithm for VGM.

1.1. Related Work

Kearns et al. (1994a) showed how to learn distributions from positive examples in the realizable setting, i.e., where the true distribution is assumed to belong to the class being learned. In the same sense as their work is similar to PAC learning Valiant (1984) of distributions, our work is like agnostic learning Kearns et al. (1994b) in which no assumption on the true distribution is made.

Generative Adversarial Networks (GANs) Goodfellow et al. (2014) are an approach for generative modeling from positive examples alone, in which a generative model is trained against a discriminator that aims to distinguish real data from generated data. In some domains, GANs have been shown to outperform other methods at generating realistic-looking examples. Several shortcomings of GANs have been observed Arora et al. (2018), and GANs are still subject to the theoretical limitations we argue are inherent to any model trained without a validity oracle.

In supervised learning, there is a rich history of learning theory with various types of queries, including membership which are not unlike our (in)validity oracle. Under various assumptions, queries have been shown to facilitate the learning of complex classes such as finite automata Angluin (1988) and DNFs Jackson (1997). See the survey of Angluin (1992) for further details. Interestingly, Feldman (2009) has shown that for agnostic learning, i.e., without making assumptions on the generating distribution, the addition of membership queries does not enhance what is learnable beyond random examples alone. Supervised learning also has a large literature around active learning, showing how the ability to query examples reduces the sample complexity of many algorithms. See the survey of Hanneke (2014). Note that the aim here is typically to save examples and not to expand what is learnable.

More sophisticated models, e.g., involving neural networks, can mitigate the invalidity problem as they often generate more realistic natural language and have even been demonstrated to generate LaTeX that nearly compiles (Karpathy, 2015) or nearly valid Wikipedia markdown. However, longer strings generated are unlikely to be valid. For example, Karpathy (2015) shows generated markdown which includes:

> ==Access to ”rap=== The current history of the BGA has been [[Vatican Oriolean Diet]], British Armenian, published in 1893. While actualistic such conditions such as the [[Style Mark Romanians]] are still nearly not the loss.

Even ignoring the mismatched quotes and equal signs, note that this example has two so-called “red links” to two pages that do not exist. Without checking, it was not obvious to us whether or not Wikipedia had pages titled *Vatican Oriolean Diet* or *Style Mark Romanians*. In some applications, one may or may not want to disallow red links. In the case that they are considered valid, one may seek a full generative model of what might plausibly occur inside of brackets, as the neural network has learned in this case. If they are disallowed, a model might memorize links it has seen but not generate new ones. A validity oracle can help the learner identify what it should avoid generating.

In practice, Kusner et al. (2017) discuss how generative models from neural networks (in particular autoencoders) often generate invalid sequences. Janz et al. (2018) learn the validity of examples output by a generative model using oracle feedback.

2. Problem Formulation

We will consider a setting where we have access to a distribution p over a (possibly infinite) set X, and let p_x be the probability mass assigned by p to each $x \in X$. For simplicity, we assume that all distributions are discrete, but our results extend naturally to continuous settings as well. Let $\text{supp}(p) \subseteq X$ denote the support of distribution p. We assume we have two types of access to p:

1. Sample access: We may draw samples $x_i \sim p$;

2. Invalidity access: We may query whether a point x_i is "invalid".

To be more precise on the second point, we assume we have access to an oracle which can answer queries to the function $\textsc{Inv} : X \to \{0, 1\}$, where $\textsc{Inv}(x) = 1$ indicates that a point is "invalid." As shorthand, we will use $\textsc{Inv}(q) = \mathbf{E}_{x \sim q}[\textsc{Inv}(x)]$. Put another way, if V is the set of valid points, then $\textsc{Inv}(q) = 1 - q(V)$. Henceforth, we find it more convenient to upper-bound invalidity rather than lower-bound validity.

For this work, we will assume that $\textsc{Inv}(x) = 0$ for all $x \in \text{supp}(p)$, i.e., $\textsc{Inv}(p) = 0$, though examples may also have $\textsc{Inv}(x) = 0$ even if $p(x) = 0$. However, we note that it would be relatively straightforward to extend our results to a more general case: Given a validity oracle and set of training examples that include a mix of valid and invalid examples, one can run the validity oracle on the training examples to create a subset of valid training examples.

Our goal is to output a distribution $\hat{q}$ with low invalidity and expected loss, for some monotone decreasing loss function $L : [0, 1] \to [0, M]$. In addition to the natural loss function $L(q_x) = \min(M, \log 1/q_x)$ mentioned earlier, a convex bounded loss is $L(q_x) = \log 1/(q_x + \exp(-M))$. For a class Q of candidate distributions q over X, we aim to solve the following problem:

$$\min_{\substack{q \in Q \\ \textsc{Inv}(q)=0}} \text{Loss}(q) = \min_{\substack{q \in Q \\ \textsc{Inv}(q)=0}} \mathbf{E}_{x \sim p}\left[L(q_x)\right].$$

Let OPT be the minimum value of this objective function, and q^* be a distribution which achieves this value. In practice we can never determine with certainty whether any $\hat{q}$ has 0 invalidity. Instead, given $\varepsilon_1, \varepsilon_2 > 0$, we want that $\text{Loss}(\hat{q}) \le OPT + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \le \varepsilon_2$.

Remark 1 *Note that given a candidate distribution $\hat{q}$ it is straightforward to check whether it satifies the loss and validity requirements, with probability $1 - \delta$, by computing the empirical loss using $O\left(\frac{1}{\varepsilon_1^2} \log(1/\delta)\right)$ samples from p and by querying the invalidity oracle $O\left(\frac{1}{\varepsilon_2} \log(1/\delta)\right)$ times using samples generated from $\hat{q}$. This observation allows us to focus on distribution learning algorithms that succeed with a constant probability as we can amplify the success probability to $1 - \delta$ by repeating the learning process $O(\log(1/\delta))$ times and checking whether the ouput is correct.*

3. Proper Learning

For ease of exposition, we begin with a canonical and simple example, where our goal is to approximate the distribution p using a uniform distribution over a two-dimensional rectangle (or, in higher dimensions, a multi-dimensional box).

Here, the goal is to find a uniform distribution q^* over a rectangle that best approximates p (i.e., minimizes some loss) while lying entirely in its valid region. We are allowed to output a uniform distribution $\hat{q}$ over a rectangle that has at least $1 - \varepsilon_2$ of its mass within the valid region. Figure 1 illustrates the target distribution q^* graphically.

3.1. Example: Uniform distributions over a Box

Let $X = \{0, 1, ..., \Delta - 1\}^d$ and assume that Q is the family of distributions that are uniform over a box, i.e. for every $q \in Q$, there exists $\boldsymbol{a}, \boldsymbol{b} \in \{0, 1, ..., \Delta - 1\}^d$ such that:

$$q_x = \frac{\mathbb{I}[\forall i \in \{1, ..., d\} : x_i \in [a_i, b_i]]}{\prod_{i=1}^{d}(b_i - a_i + 1)}$$

Theorem 2 *Using $O\left(\frac{dM^2}{\varepsilon_1^2}\right)$ samples and $\frac{1}{\varepsilon_2}\left(\frac{dM}{\varepsilon_1}\right)^{O(d)}$ invalidity queries on p, there exists an algorithm which identifies a distribution $\hat{q} \in Q$, such that $\textsc{Inv}(\hat{q}) \le \varepsilon_2$ and $Loss(\hat{q}) \le Loss(q^*) + \varepsilon_1$ with probability $3/4$.*

Proof Since the VC-dimension of d-dimensional boxes is $2d$, with probability $7/8$ after taking a set X_P of $P = O\left(\frac{dM^2}{\varepsilon_1^2}\right)$ samples from p, we can estimate $p(\text{supp}(q))$ for all distributions $q \in Q$ within $\pm\frac{\varepsilon_1}{2M}$ by forming the empirical distribution. This implies that the empirical loss $\overline{\text{Loss}}(q) = \frac{1}{|X_P|}\sum_{x \in X_p} L(q_x)$ is an estimate to the loss function, i.e. $\overline{\text{Loss}}(q) \in \text{Loss}(q) \pm \frac{\varepsilon_1}{2}$.

Now consider the optimal distribution q^*. Observe that any distribution $q \in Q$, such that $\text{supp}(q) \subseteq \text{supp}(q^*)$ and $\text{supp}(q) \cap X_P = \text{supp}(q^*) \cap X_P$, satisfies $\overline{\text{Loss}}(q) \le \overline{\text{Loss}}(q^*)$ and $\textsc{Inv}(q) = 0$. Thus, there exists a $q' \in Q$ with this property that has at least one point $x \in X_P$ in each of the $2d$ sides of its box.

As there are at most P^{2d} such boxes, we can check identify which of their corresponding distribution $q \in Q$ have $\textsc{Inv}(q) \le \varepsilon_2$ by quering $\textsc{Inv}$ at $O\left(\frac{1}{\varepsilon_2}\log\left(P^{2d}\right)\right)$ random points from each of them. This succeeds with probability $7/8$ and uses in total $\frac{1}{\varepsilon_2}\left(\frac{dM}{\varepsilon_1}\right)^{O(d)}$ invalidity queries.

We pick $\hat{q}$ to be the distribution that minimizes the empirical $\overline{\text{Loss}}(\hat{q})$ out of those that have no invalid samples in the support. Overall, with probability $3/4$, we have that $\textsc{Inv}(\hat{q}) \le \varepsilon_2$ and

$$\text{Loss}(\hat{q}) \le \overline{\text{Loss}}(\hat{q}) + \frac{\varepsilon_1}{2} \le \overline{\text{Loss}}(q') + \frac{\varepsilon_1}{2} \le \overline{\text{Loss}}(q^*) + \frac{\varepsilon_1}{2} \le \text{Loss}(q^*) + \varepsilon_1.$$

■

3.2. Impossibility of Proper Learning

The example in the previous section required number of queries that is exponential in d in order to output a distribution $\hat{q} \in Q$ with $\textsc{Inv}(\hat{q}) \le \varepsilon_2$ and $\text{Loss}(\hat{q}) \le \text{Loss}(q^*) + \varepsilon_1$. We show that such an exponential dependence in d is required when one aims to learn a distribution $\hat{q}$ properly even for the class of uniform distributions over axis-parallel boxes.

The proof of the following theorem appears in Section A:

Theorem 3 *Even for $\Delta = 2$, the number of queries required to find a distribution $\hat{q} \in Q$ such that $\textsc{Inv}(\hat{q}) \le \frac{1}{4}$ and $Loss(\hat{q}) \le Loss(q^*) + \frac{1}{2d}$ with probability at least $3/4$ is at least $2^{\Omega(d)}$.*

As Theorem 3 shows, proper learning suffers from a “needle in a haystack” phenomenon. To build intuition, we present an alternative simpler setting that illustrates this point more clearly.

Let Q be the set of all distributions q_i that, with probability $\frac{1}{2}$, output 0, and otherwise output $i > 0$. Let p be the distribution that always outputs 0 and suppose that $\textsc{Inv}(i) = 1$ for all $i \neq \{0, i^*\}$

for some arbitrary i^*. In order to properly learn the distribution $\hat{q}$, one needs to locate the hidden i^* by querying the invalidity oracle many times. This requires a number of queries that is proportional to the size of the domain X, which is intractable when the domain is large (e.g., in high dimensions) or even infinite.

Note, however, that in this example, even though learning a distribution q within the family Q is hard, we can easily come up with an improper distribution that always outputs point 0. Such a distribution is always valid and achieves optimal loss. In the next section we show that even though proper learning may be information-theoretically expensive or impossible, it is actually always possible to improperly learn using polynomially many samples and invalidity queries.

4. Improper Learning

In this section, we show that if we are allowed to output a distribution that is not in the original family Q, we can efficiently identify a distribution that achieves close to optimal loss and almost-full validity using only polynomially many samples from p and invalidity queries.

4.1. Algorithm

We provide an algorithm, Algorithm 1, that can solve the task computationally efficiently assuming access to an optimization oracle Oracle(X_P, X_N). Oracle(X_P, X_N) takes as input sets X_P and X_N of positive and negative (invalid) points and outputs a distribution q from the family of distributions Q that minimizes the empirical loss with respect to X_P such that supp$(q) \cap X_N = \emptyset$, i.e. no negative point in X_N is in the support of q.

Algorithm 1: Improperly learning to generate valid samples

```
Input: Distribution family $Q$, sample and invalidity access to $p$, and parameters $\varepsilon_1, \varepsilon_2 > 0$.
Draw a set $X_P$ of $P$ samples from $p$.
Set $X_N \leftarrow \emptyset$
for $i = 1, ..., R$ do
    Let $q^i \leftarrow$ Oracle$(X_P, X_N)$.
    Generate $T$ samples from $q^i$ and query the invalidity of each of them.
    Let $x_1^-, ..., x_k^-$ be the invalid samples.
    if there are no invalid samples, i.e. $k = 0$ then
        return $q^i$
    else
        Set $X_N \leftarrow X_N \cup \{x_1^-, ..., x_k^-\}$
    end if
end for
Sample $i \sim$ Uniform$(\{1, ..., R\})$
Let $A^i \leftarrow \{x : \exists j > i$ with $x \in$ supp$(q^j)\}$
return the distribution that samples $x \sim q^i$ and outputs $x$ if $x \in A^i$ and any valid point $x^*$ o/w
```

The algorithm repeatedly finds the distribution with minimum loss that doesn't contain any of the invalid points seen so far and tests whether it achieves almost full-validity. If it does, then it outputs that distribution. Otherwise it tries again using the new set of invalid points. However, this process could repeat for a very long time without finding a distribution. To avoid this, after running for a few rounds, if it has failed to output a distribution, the algorithm is able to generate an improper distribution that provides the required guarantee to solve the task. This meta-distribution is obtained by randomly picking one of the candidate distributions examined so far and filtering out points that no other distributions agree on.

4.2. Analysis

We show that this Algorithm 1 outputs with high probability a distribution $\hat{q}$ that has $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \leq \varepsilon_2$.

Theorem 4 *The choice of parameters*

$$P = \Theta\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right), \quad R = \Theta\left(\frac{M}{\varepsilon_1}\right), \quad T = \Theta\left(\frac{R}{\varepsilon_2}\log|Q|\right) \tag{1}$$

guarantees that Algorithm 1 outputs w.p. $3/4$ *a distribution* $\hat{q}$ *with* $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$ *and* $\textsc{Inv}(\hat{q}) \leq \varepsilon_2$ *using* $\Theta\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right)$ *samples from p and* $\Theta\left(\frac{M^2}{\varepsilon_1^2\varepsilon_2}\log|Q|\right)$ *invalidity queries.*

The algorithm runs in time polynomial in M, ε_1^{-1}, ε_2^{-1}, *and* $\log|Q|$ *assuming that the following each can be performed at unit cost: (a) queries to Oracle, (b) sampling from the distributions output by Oracle, and (c) checking whether a point* x *is in the support of a distribution output by Oracle.*

Of course, the success probability can be boosted from 3/4 to arbitrarily close to $1-\delta$ by repeating the algorithm $O(\log 1/\delta)$ times and taking the best output. We prove Theorem 4 by showing two lemmas, Lemma 5 and Lemma 6, bounding the invalidity and loss of the returned distribution.

Lemma 5 *The returned distribution* $\hat{q}$ *by Algorithm 1 satisfies* $\textsc{Inv}(\hat{q}) \leq \varepsilon_2$ *w.p.* $7/8$.

Proof Let Invalid $= \{x : \textsc{Inv}(x) = 1\}$ be the set of invalid points. Consider q^i for some i and any distribution $q \in Q$. If $q^i(\text{supp}(q) \cap \text{Invalid}) \geq \frac{\varepsilon_2}{R}$, then with probability at least $\frac{\varepsilon_2}{R}$ a sample generated from q^i lies in $\text{supp}(q) \cap \text{Invalid}$. Thus, with $T = \Theta(\frac{R}{\varepsilon_2}\log|Q|)$ samples at least one lies in $\text{supp}(q) \cap \text{Invalid}$ w.p. $1 - \frac{1}{8|Q|R}$. By a union bound for all i and $q \in Q$, we get that with probability $7/8$ for all q_i and all distributions $q \in Q$, if $q^i(\text{supp}(q) \cap \text{Invalid}) \geq \frac{\varepsilon_2}{R}$ then at least one of the T samples drawn from q^i lies in $\text{supp}(q) \cap \text{Invalid}$. We therefore assume that this holds.

Then, if the returned distribution $\hat{q} = q^i$ for some i, we get

$$\textsc{Inv}(q^i) = q^i(\text{supp}(q^i) \cap \text{Invalid}) < \frac{\varepsilon_2}{R} \leq \varepsilon_2$$

as required. To complete the proof we show the required property when returned distribution $\hat{q}$ is the improper meta-distribution.

We have that for all $j > i$, $q^i(\text{supp}(q^j) \cap \text{Invalid}) < \frac{\varepsilon_2}{R}$ since after round i for any $q \in Q$ with $q^i(\text{supp}(q) \cap \text{Invalid}) \geq \frac{\varepsilon_2}{R}$ the set X_N will contain at least one point in $\text{supp}(q) \cap \text{Invalid}$ and thus any such q will not be considered.

Therefore, we have that

$$\textsc{Inv}(\hat{q}) = \mathbf{E}_{x\sim\hat{q}}\left[\textsc{Inv}(x)\right] = \mathbf{E}_{x\sim q^i}\left[\textsc{Inv}(x) \cdot \mathbb{I}\left[\exists j > i : x \in \text{supp}(q^j)\right]\right]$$
$$\leq \sum_{j=i+1}^{R} \mathbf{E}_{x\sim q^i}\left[\textsc{Inv}(x) \cdot \mathbb{I}\left[x \in \text{supp}(q^j)\right]\right] = \sum_{j=i+1}^{R} q^i(\text{supp}(q^j) \cap \text{Invalid}) \leq \sum_{j=i+1}^{R} \frac{\varepsilon_2}{R} < \varepsilon_2.$$

■

Lemma 6 *The returned distribution $\hat{q}$ by Algorithm 1 satisfies Loss$(\hat{q}) \leq$ Loss$(q^*) + \varepsilon_1$ w.p. $7/8$.*

Proof Since we draw $P = \Theta\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right)$ samples from p, we have that the empirical loss $\overline{\text{Loss}}(q) \in \text{Loss}(q) \pm \frac{\varepsilon_1}{4}$ for all $q \in Q$ with probability $1 - 1/16$. We thus assume from here on that this is true.

In that case, must be that $\overline{\text{Loss}}(q^i) \leq \overline{\text{Loss}}(q^*)$. This is because the algorithm terminates if $q^i = q^*$ since q^* generates no invalid samples and no q^i with $\overline{\text{Loss}}(q^i) > \overline{\text{Loss}}(q^*)$ will be considered before examining q^*.

This implies that at any point, we have that $\text{Loss}(q^i) \leq \overline{\text{Loss}}(q^i) + \frac{\varepsilon_1}{4} \leq \overline{\text{Loss}}(q^*) + \frac{\varepsilon_1}{4} \leq \text{Loss}(q^*) + \frac{\varepsilon_1}{2}$.

Therefore, in the case that the distribution that is output is $\hat{q} = q^i$ it will satisfy the given condition. To complete the proof we show the required property when returned distribution $\hat{q}$ is the improper meta-distribution.

In that case, we have that for any $i \in [R]$:

$$\begin{aligned}
\text{Loss}(\hat{q}) &\leq \mathbf{E}_{x\sim p}\left[L\left(q^i_x \cdot \mathbb{I}\left[\exists j > i : x \in \text{supp}(q^j)\right]\right)\right] \\
&\leq \text{Loss}(q^i) + M \cdot \Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \\
&\leq \text{Loss}(q^*) + \frac{\varepsilon_1}{2} + M \cdot \Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]
\end{aligned}$$

However, since a random index $i \sim \text{Uniform}(\{1, ..., R\})$ is chosen, we have that in expectation over this random choice

$$\begin{aligned}
&\mathbf{E}_i\left[\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]\right] \\
&\leq \frac{1}{R}\sum_{i=1}^{R}\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \\
&\leq \frac{1}{R}\mathbf{E}_{x\sim p}\left[\sum_{i=1}^{R}\mathbb{I}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]\right] \leq \frac{1}{R}
\end{aligned}$$

where the last inequality follows since $\sum_{i=1}^{R}\mathbb{I}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \leq 1$ as only the largest i with $x \in \text{supp}(q^i)$ has that for all $j > i$, $x \notin \text{supp}(q^j)$.

By Markov's inequality, we have that with probability $1 - 1/16$, a random i will have

$$\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \leq \frac{16}{R}.$$

Therefore, the choice of $R = 32\frac{M}{\varepsilon_1} = \Theta\left(\frac{M}{\varepsilon_1}\right)$ guarantees that $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$. The overall failure probability is at most $1/16 + 1/16 = 1/8$.

■

5. Extensions

5.1. Partial validity

In this section, we consider a generalization of our main setting, where we allow some slack in the validity constraint. More precisely, given some parameter $\alpha > 0$, we now have the requirement that $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \leq \alpha + \varepsilon_2$, where q^* is the optimal distribution which minimizes $\text{Loss}(q^*)$ such that $\textsc{Inv}(q^*) \leq \alpha$.

5.1.1. ALGORITHM

We provide an algorithm for solving the partial validity problem in Algorithm 2. This method is sample-efficient, requiring a number of samples which is $\text{poly}\left(M, \varepsilon_1^{-1}, \varepsilon_2^{-1}, \log |Q|\right)$.

Algorithm 2: Learning a distribution with partial validity

```
Input: Sample and invalidity access to a distribution $p$, parameters $\varepsilon_1, \varepsilon_2, \alpha > 0$, a family of distributions $Q$.
Using $n_1$ samples from $p$, empirically estimate $\overline{\text{Loss}}(q) \in \text{Loss}(q) \pm \frac{\varepsilon_1}{3}$ for all $q \in Q$.
for $\ell \in \left\{0, \frac{\varepsilon_1}{3}, ..., M\right\}$ do
  Let $D = \{q \in Q \mid \overline{\text{Loss}}(q) \leq \ell\}$.
  Let $x^*$ be any point with $\textsc{Inv}(x^*) = 0$.
  Let $\mu_D$ be the distribution which samples a distribution $q$ uniformly from $D$, and then draws a sample from $q$.
  while $D \neq \emptyset$ do
    Draw $n_2$ samples $x_1, ..., x_{n_2}$ from $\mu_D$.
    if $\frac{1}{n_2} \sum_{i=1}^{n_2} \textsc{Inv}(x_i) \Pr_{q \sim \text{Uniform}(D)}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M] \leq \alpha + \frac{4\varepsilon_2}{5}$ then
      return $\mu'_D$, which samples $x$ from $\mu_D$ with probability
          $\Pr_{q \sim \text{Uniform}(D)}[q(x)\varepsilon_1 < 3\mu_D(x)M],$
      and samples $x^*$ otherwise.
    else
      Remove all distributions $q$ from $D$ for which
          $\frac{1}{n_2} \sum_{i=1}^{n_2} \textsc{Inv}(x_i) \frac{q(x_i)}{\mu_D(x_i)} \mathbb{I}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M] > \alpha + \frac{\varepsilon_2}{5}.$
    end if
  end while
end for
```

5.1.2. ANALYSIS

We will show that, with high probability, Algorithm 2 outputs a distribution $\hat{q}$ that has $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \leq \alpha + \varepsilon_2$.

Theorem 7 *Suppose that the loss function L is convex. The choice of parameters*

$$n_1 = \Theta\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right), n_2 = \Theta\left(\frac{M^2}{\varepsilon_1^2\varepsilon_2^2}\log|Q|\log\left(\frac{M\log|Q|}{\varepsilon_1\varepsilon_2}\right)\right) \tag{2}$$

guarantees that Algorithm 2 outputs w.p. $3/4$ a distribution with Loss$(\hat{q}) \leq$ Loss$(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \leq \alpha + \varepsilon_2$ using $\Theta\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right)$ samples from p and $\Theta\left(\frac{M^3}{\varepsilon_1^3\varepsilon_2^3}\log^2|Q|\log\left(\frac{M\log|Q|}{\varepsilon_1\varepsilon_2}\right)\right)$ invalidity queries.

Remark 8 *We note that this algorithm still works in the case where points may be "partially valid" – specifically, we let $\textsc{Inv} : X \to [0,1]$ take fractional values. This requires that we have access to some point x^* where $\textsc{Inv}(x^*) = 0$, which we assume is given to us by some oracle. For instance, the distribution may choose to output a dummy symbol $\perp$, rather than output something which may not be valid.*

We prove Theorem 7 through three lemmas. The sample complexity bound follows from the values of n_1, n_2, the fact that we have at most $O\left(\frac{M}{\varepsilon_1}\right)$ iterations of the loop at Line 3, and Lemma 9 which bounds the number of iterations of the loop at Line 7 as $O\left(\frac{\log|Q|}{\varepsilon_2}\right)$ for any ℓ. To argue correctness, Lemmas 10 and 11 bound the invalidity and loss of any output distribution, respectively. The proofs of these lemmata appear in Section B.

Lemma 9 *With probability at least $14/15$, the loop at Line 7 requires at most $O\left(\frac{\log|Q|}{\varepsilon_2}\right)$ iterations for each ℓ.*

Lemma 10 *With probability at least $14/15$, if at any step a distribution μ'_D is output, $\textsc{Inv}(\mu'_D) \leq \alpha + \varepsilon_2$.*

Lemma 11 *With probability at least $14/15$, if at any step a distribution μ'_D is output, Loss$(\mu'_D) \leq \ell + 2\varepsilon_1/3$, where ℓ is the step at which the distribution was output.*

The proof of Theorem 7 concludes by observing that the optimal distribution q^* is never eliminated (assuming all estimates involving its loss and validity are accurate, which happens with probability at least $19/20$), and that the loop in line 3 steps by increments of $\varepsilon_1/3$. Combining this with Lemma 11, if we output $\hat{q}$, then $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$.

5.2. General Densities

For simplicity of presentation, we have formulated the above results in terms of probability mass functions q on a discrete domain X. However, we note that all of the above results easily extend to general density functions on an abstract measurable space X, which may be either discrete or uncountable. Specifically, if we let μ_0 denote an arbitrary reference measure on X, then we may consider the family Q to be a set of *probability density functions* q with respect to μ_0: that is, non-negative measurable functions such that $\int q\mathrm{d}\mu_0 = 1$. For the results above, we require that we have a way to (efficiently) generate iid samples having the distribution whose density is q. For the full-validity results, the only additional requirements are that we are able to (efficiently) test whether a given x is in the support of q, and that we have access to Oracle$(\cdot,\cdot)$ defined with respect to the set

Q. For the results on partial-validity, we require the ability to explicitly evaluate the function q at any $x \in X$. The results then hold as stated, and the proofs remain unchanged (overloading notation to let q_x denote the value of the density q at x, and $q(A) = \int_A q \mathrm{d}\mu_0$ the measure of A under the probability measure whose density is q).

5.3. Infinite Families of Distributions

It is also possible to extend all of the above results to *infinite* families Q, expressing the sample complexity requirements in terms of the *VC dimension* (Vapnik and Chervonenkis (1974)) of the supports $d = \mathrm{VCdim}(\{\mathrm{supp}(q) : q \in Q\})$, and the *fat-shattering dimension* (Alon et al. (1997)) of the family of loss-composed densities $s(\varepsilon) = \mathrm{fat}_\varepsilon(\{x \mapsto L(q_x) : q \in Q\})$. In this case, in the context of the full-validity results, for simplicity we assume that in the evaluations of $\mathrm{Oracle}(X_P, X_N)$ defined above, there always *exists* at least one minimizer $q \in Q$ of the empirical loss with respect to X_P such that $\mathrm{supp}(q) \cap X_N = \emptyset$.[2] We then have the following result. For completeness, we include a full proof in the appendix.

Theorem 12 *For a numerical constant $c \in (0, 1]$, the choice of parameters*

$$P = \Theta\left(\frac{s(c\varepsilon_1/M)M^2}{\varepsilon_1^2} \log \frac{M}{\varepsilon_1}\right), \quad R = \Theta\left(\tfrac{M}{\varepsilon_1}\right), \quad T = \Theta\left(\frac{Rd}{\varepsilon_2} \log \frac{1}{\varepsilon_2}\right)$$

guarantees that Algorithm 1 outputs w.p. $3/4$ a distribution $\hat{q}$ with Loss$(\hat{q}) \le$ Loss$(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \le \varepsilon_2$ *using P samples from p and RT invalidity queries.*

The algorithm runs in time polynomial in M, ε_1^{-1}, ε_2^{-1}, d, and $s_{\varepsilon_1/256}$ assuming that queries to the optimization oracle can be computed in polynomial time. Moreover, sampling from the resulting distribution $\hat{q}$ can also be performed in polynomial time.

For partial-validity, we can also extend to infinite Q, though in this case via a more-cumbersome technique. Specifically, let us suppose the densities $q \in Q$ are bounded by 1 (this can be replaced by any value by varying the sample size n_2). Then we consider running Algorithm 2 as usual, except replacing Step 4 with the step

$$D = \mathrm{Cover}_{\varepsilon_2}(\{q \in Q | \overline{\mathrm{Loss}}(q) \le \ell\}),$$

where for any $R \subseteq Q$, $\mathrm{Cover}_{\varepsilon_2}(R)$ denotes a minimal subset of R such that $\forall q \in R$, $\exists q^{\varepsilon_2} \in \mathrm{Cover}_{\varepsilon_2}(R)$ with $\int |q_x - q_x^{\varepsilon_2}| \mu_0(\mathrm{d}x) \le \varepsilon_2$: that is, an ε_2-cover of R under $L_1(\mu_0)$. Let us refer to this modified algorithm as Algorithm $2'$. We have the following result.

Theorem 13 *Suppose that the loss function L is convex. For a numerical constant $c \in (0, 1]$, the choice of parameters*

$$n_1 = \Theta\left(\frac{s(c\varepsilon_1/M)M^2}{\varepsilon_1^2} \log\left(\frac{M}{\varepsilon_1}\right)\right), \quad n_2 = \Theta\left(\frac{M^2 \mathrm{fat}_{c\varepsilon_2}(Q)}{\varepsilon_1^2 \varepsilon_2^2} \log^2\left(\frac{M \mathrm{fat}_{c\varepsilon_2}(Q)}{\varepsilon_1 \varepsilon_2}\right)\right)$$

guarantees that Algorithm $2'$ (with parameters ε_1, ε_2, and $\alpha + \varepsilon_2$) outputs w.p. $3/4$ a distribution with Loss$(\hat{q}) \le$ Loss$(q^*) + \varepsilon_1$ and $\textsc{Inv}(\hat{q}) \le \alpha + 2\varepsilon_2$ *using n_1 samples from p and* $\Theta\left(\frac{M^3 \mathrm{fat}_{c\varepsilon_2}(Q)^2}{\varepsilon_1^3 \varepsilon_2^3} \log^3\left(\frac{M \mathrm{fat}_{c\varepsilon_2}(Q)}{\varepsilon_1 \varepsilon_2}\right)\right)$ *invalidity queries.*

[2] It is straightforward to remove this assumption by supposing $\mathrm{Oracle}(X_P, X_N)$ returns a q that *very-nearly* minimizes the empirical loss, and handling this case requires only superficial modifications to the arguments.

References

Noga Alon, Shai Ben-David, Nicolo Cesa-Bianchi, and David Haussler. Scale-sensitive dimensions, uniform convergence, and learnability. *Journal of the ACM*, 44(4):615–631, 1997.

Dana Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

Dana Angluin. Computational learning theory: Survey and selected bibliography. In *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, STOC '92, pages 351–369, New York, NY, USA, 1992. ACM.

Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018.

Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K. Warmuth. Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM*, 36(4):929–965, 1989.

Vitaly Feldman. On the power of membership queries in agnostic learning. *Journal of Machine Learning Research*, 10(Feb):163–182, 2009.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems 27*, NIPS '14, pages 2672–2680. Curran Associates, Inc., 2014.

Steve Hanneke. Theory of disagreement-based active learning. *Foundations and Trends® in Machine Learning*, 7(2–3):131–309, 2014.

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

Jeffrey C. Jackson. An efficient membership-query algorithm for learning DNF with respect to the uniform distribution. *Journal of Computer and System Sciences*, 55(3):414–440, 1997.

David Janz, Jos van der Westhuizen, Brooks Paige, Matt J. Kusner, and José Miguel Hernández-Lobato. Learning a generative model for validity in complex discrete structures. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR '18, 2018.

Dan Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Prentice Hall, 2009.

Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. `http://karpathy.github.io/2015/05/21/rnn-effectiveness/`, May 2015.

Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. On the learnability of discrete distributions. In *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, STOC '94, pages 273–282, New York, NY, USA, 1994a. ACM.

Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Towards efficient agnostic learning. *Machine Learning*, 17(2–3):115–141, 1994b.

Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar variational autoencoder. In *Proceedings of the 34th International Conference on Machine Learning*, ICML '17, pages 1945–1954. JMLR, Inc., 2017.

Cyril Labbé and Dominique Labbé. Duplicate and fake publications in the scientific literature: How many SCIgen papers in computer science? *Scientometrics*, 94(1):379–396, 2013.

Shahar Mendelson and Roman Vershynin. Entropy and the combinatorial dimension. *Inventiones Mathematicae*, 152(1):37–55, 2003.

Ronald Rosenfeld. Two decades of statistical language modeling: Where do we go from here? *Proceedings of the IEEE*, 88(8):1270–1278, 2000.

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

Vladimir Vapnik and Alexey Chervonenkis. *Theory of Pattern Recognition*. Nauka, 1974.

Ian H. Witten, Radford M. Neal, and John G. Cleary. Arithmetic coding for data compression. *Communications of the ACM*, 30(6):520–540, 1987.

Appendix A. Proof of Theorem 3

We describe the construction of the lower-bound below:

- The distribution p assigns probability $1/d$ to each standard basis vector $\boldsymbol{e_i}$, i.e., the vector with i-th entry equal to 1 and all other coordinates equal to 0.
- For some arbitrary vector $y \in \{0,1\}^d$ with $|y| = \sum_{i=1}^d y_i = d/3$, we define $\textsc{Inv}(x)$ as:

$$\textsc{Inv}(x) = \begin{cases} 0 & \textbf{if } |x| < d/6 \textbf{ or } \text{for all } i, x_i \le y_i \\ 1 & \text{otherwise.} \end{cases}$$

- The loss function is the coverage function, i.e., $L(q_x) = \mathbb{I}[q_x = 0]$, where we pay a loss of 1 for each point q assigns 0 mass to, and 0 otherwise.

Given this instance, the optimal q^* is uniform over the box $\times_{i=1}^d \{0, y_i\}$ and has loss $\frac{2}{3}$. In order to achieve loss $\frac{2}{3} + \frac{1}{2d}$, the output distribution $\hat{q}$ must include at least $d/3$ of the vectors $\boldsymbol{e}_i$ in its support. Thus, $\hat{q}$ must be a box $\times_{i=1}^d \{0, y'_i\}$ defined by some vector $y' \in \{0,1\}^d$ with $|y'| \ge d/3$. Moreover, it must be that $y' = y$. This is because if there exists a coordinate j such that $y'_j = 1$ and $y_j = 0$, then with probability greater than $1/4$, the distribution q produces a sample x with $x_j = 1$ and $|x| \ge d/6$. Since such a sample is invalid, $\textsc{Inv}(\hat{q}) > \frac{1}{4}$ which would lead to a contradiction.

Therefore the goal is to find the vector y. Since any samples from p only produce points e_i they provide no information about y. Furthermore, queries to $\textsc{Inv}$ at points x with $|x| < d/6$ or $|x| > d/3$ also provide no information about y, as in the former case $\textsc{Inv}(x) = 0$ since $|x| < d/6$, and in the latter case $\textsc{Inv}(x) = 1$ since there will always be an i where $1 = x_i > y_i = 0$. Therefore, it only makes sense to query points with $|x| \in [d/6, d/3]$.

We show that the number of queries needed to identify the true y is exponential in d. We do this with a Gilbert-Varshamov style argument. To see this, consider a set of vectors $Y \subset \{0,1\}^d$ such that for all $y' \in Y$ we have that $|y'| = d/3$ and any two distinct vectors $y^1, y^2 \in Y$ have fewer than $d/6$ coordinates where they are both 1, i.e. $\sum_i y_i^1 \cdot y_i^2 < d/6$.

Given this set Y, note that any query to $\textsc{Inv}$ at a point x with $|x| \in [d/6, d/3]$ eliminates at most a single $y' \in Y$. Thus with fewer than $|Y|/2$ queries, the probability that the true y is identified is less than $1/2$.

To complete the proof, we show that a set Y exists with $|Y| = e^{d/216}$. We will use a randomized construction where we pick $|Y|$ random points $y^1, ..., y^{|Y|} \in \{0,1\}^d$ with $|y^a| = d/3$ uniformly at random. Consider two such random points y^a and y^b.

Define the random variable z_i to be 1 if $y_i^1 = y_i^2 = 1$ and 0 otherwise. We have

$$\Pr[z_i = 1] = \frac{1}{3} \cdot \frac{1}{3} = \frac{1}{9}.$$

Although z_i's are not independent, they are negative correlated. We can apply the multiplicative Chernoff bound:

$$\Pr\left[\sum_{i=1}^{d} z_i \geq d/6\right] \leq e^{-d/108}$$

Then by a union bound over all pairs $a < b$, we have

$$\Pr[\forall 1 \leq a < b \leq |Y|, \sum_i y_i^a \cdot y_i^b < d/6] > 1 - \binom{|Y|}{2} \cdot e^{-d/108} > 0.$$

This shows that the number of queries an algorithm must make to succeed with probability at least $3/4$ is at least $2^{\Omega(d)}$.

Appendix B. Missing Proofs from Section 5.1

B.1. Proof of Lemma 9

To bound the number of iterations, we will show that if no distribution is output, $|D|$ shrinks by a factor $1 - \frac{\varepsilon_2}{5}$. As we start with at most $|Q|$ candidate distributions, this implies the required bound.

We note that we have a multiplicative term $\log\left(\frac{M \log |Q|}{\varepsilon_1 \varepsilon_2}\right)$ in the expression for n_2. This corresponds to certain estimates being accurate for the first $\mathrm{poly}(M, \log |Q|, \varepsilon_1^{-1}, \varepsilon_2^{-1})$ times they are required by a union bound argument. As this proof will justify, each line in the algorithm is run at most $\frac{M \log |Q|}{\varepsilon_1 \varepsilon_2}$ times. Thus, for ease of exposition, we simply will state that estimates are accurate for every time the line is run.

We thus need to count how many candidate distributions in D are eliminated in every round given that the empirical invalidity of μ'_D is at least $\alpha + \frac{4\varepsilon_2}{5}$, i.e.

$$\frac{1}{N} \sum_{i=1}^{N} \textsc{Inv}(x_i) \Pr_{q \sim \text{Uniform}(D)}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M] > \alpha + \frac{4\varepsilon_2}{5}.$$

This implies that the true invalidity of μ'_D is at least $\alpha + \frac{3\varepsilon_2}{5}$: since $n_2 = \Omega\left(\frac{1}{\varepsilon_2^2} \cdot \log\left(\frac{M \log |Q|}{\varepsilon_1 \varepsilon_2}\right)\right)$, we have that $\overline{\textsc{Inv}}(\mu'_D) = \textsc{Inv}(\mu'_D) \pm \frac{\varepsilon_2}{5}$ each time this line is run, with probability $29/30$.

Similarly, for every q we have that the estimator $\frac{1}{n_2}\sum_{i=1}^{n_2}\textsc{Inv}(x_i)\frac{q(x_i)}{\mu_D(x_i)}\mathbb{I}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M]$ is an accurate estimator for the validity of q' which is the distribution that generates a sample x from q and returns x if $q(x)\varepsilon_1 \leq 3\mu_D(x)M$ and x^* otherwise. This is because, since $\textsc{Inv}(x^*) = 0$, we have

$$\begin{aligned}\mathbf{E}_{x\sim\mu_D}\left[\textsc{Inv}(x)\frac{q(x)}{\mu_D(x)}\mathbb{I}[q(x)\varepsilon_1 < 3\mu_D(x)M]\right] &= \mathbf{E}_{x\sim q}\left[\textsc{Inv}(x)\mathbb{I}[q(x)\varepsilon_1 < 3\mu_D(x)M]\right] \\ &= \mathbf{E}_{x\sim q'}\left[\textsc{Inv}(x)\right] = \textsc{Inv}(q').\end{aligned}$$

Note that our estimate $\overline{\textsc{Inv}}(q')$ is the empirical value

$$\frac{1}{n_2}\sum_{i=1}^{n_2}\textsc{Inv}(x_i)\frac{q(x_i)}{\mu_D(x_i)}\mathbb{I}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M],$$

where

$$\frac{q(x_i)}{\mu_D(x_i)}\mathbb{I}[q(x_i)\varepsilon_1 < 3\mu_D(x_i)M] \leq \frac{3M}{\varepsilon_1}.$$

Since we are estimating the expectation of a function upper bounded by $O(M/\varepsilon_1)$ and there are at most $|Q|$ distributions q' at each iterations, $n_2 = \Omega\left(\frac{M^2}{\varepsilon_1^2\varepsilon_2^2}\log|Q|\log\left(\frac{M\log|Q|}{\varepsilon_1\varepsilon_2}\right)\right)$ samples are sufficient to have that the empirical estimator $\overline{\textsc{Inv}}(q') = \textsc{Inv}(q') \pm \frac{\varepsilon_2}{5}$ for all distributions q' considered and all times this line is run, with probability $29/30$. Thus, it is sufficient to count how many $q \in D$ exist with $\textsc{Inv}(q') > \alpha + \frac{3\varepsilon_2}{5}$.

To do this, we notice that $\mathbf{E}_{q\in\text{Uniform}(D)}[\textsc{Inv}(q')] = \textsc{Inv}(\mu'_D) > \alpha + \frac{3\varepsilon_2}{5}$. Then, as $\textsc{Inv}(q') \leq 1$, we have that $\Pr_{q\sim\text{Uniform}(D)}[\textsc{Inv}(q') > \alpha + \frac{2\varepsilon_2}{5}] \geq \frac{\varepsilon_2}{5}$. This yields the required shrinkage of the set D.

B.2. Proof of Lemma 10

The estimator $\frac{1}{n_2}\sum_{i=1}^{n_2}\textsc{Inv}(x_i)\Pr_{q\sim\text{Uniform}(D)}[q(x_i)\varepsilon_1 < 2\mu_D(x_i)M]$ estimates the empirical fraction of samples that are invalid for distribution μ'_D. Since $n_2 = \Omega\left(\frac{1}{\varepsilon_2^2}\log\left(\frac{M\log|Q|}{\varepsilon_1\varepsilon_2}\right)\right)$, and by Lemma 9 each line is run at most $O\left(\frac{M\log|Q|}{\varepsilon_1\varepsilon_2}\right)$ times, the empirical estimate of $\overline{\textsc{Inv}}(\mu'_D) = \textsc{Inv}(\mu'_D) \pm \frac{\varepsilon_2}{5}$ for all iterations, with probability at least $14/15$. The statement holds as μ'_D is only returned if the estimate for the invalidity of μ'_D is at most $\alpha + \frac{4\varepsilon_2}{5}$.

B.3. Proof of Lemma 11

For any $q \in D$ denote by q' the distribution that generates a sample x from q and returns x if $q(x)\varepsilon_1 \leq 3\mu_D(x)M$ and x^* otherwise. Notice that $\mu'_D(x) = \mathbf{E}_{q\sim\text{Uniform}(D)}[q'(x)]$. We have that

$$\begin{aligned}\text{Loss}(\mu'_D) &= \mathbf{E}_{x\sim p}[L(\mu'_D(x))] \leq \mathbf{E}_{x\sim p}[\mathbf{E}_{q\sim\text{Uniform}(D)}[L(q'(x))]] \\ &\leq \mathop{\mathbf{E}}_{\substack{x\sim p\\ q\sim\text{Uniform}(D)}}[L(q(x)) + M\cdot\mathbb{I}[q(x)\varepsilon_1 > 3\mu_D(x)M]] \\ &\leq \sup_{q\in D}\text{Loss}(q) + M\cdot\Pr_{\substack{x\sim p\\ q\sim\text{Uniform}(D)}}[q(x)\varepsilon_1 > 3\mu_D(x)M]\end{aligned}$$

The equality is the definition of Loss, the first inequality uses convexity of L and Jensen's inequality, and the second inequality uses the fact that $L(\cdot) \leq M$.

However, for any given x, we have that $\mathbf{E}_{q\sim\text{Uniform}(D)}[q(x)] = \mu_D(x)$ and thus by Markov's inequality we obtain that for all x

$$\Pr_{q\sim\text{Uniform}(D)}[q(x)\varepsilon_1 > 3\mu_D(x)M] \leq \frac{\varepsilon_1}{3M}.$$

This implies that $M \cdot \Pr_{\substack{x\sim p\\ q\sim\text{Uniform}(D)}}[q(x)\varepsilon_1 > 3\mu_D(x)M]$ is at most $\frac{\varepsilon_1}{3}$. To complete the proof we note that $\sup_{q\in D}\text{Loss}(q)$ is at most $\ell + \frac{\varepsilon_1}{3}$: since we are estimating the mean of $L(\cdot)$ which is bounded by M, there are $|Q|$ distributions q which are considered, and $n_1 = \Omega\left(\frac{M^2}{\varepsilon_1^2}\log|Q|\right)$, the statement holds for all q simultaneously with probability at least $14/15$.

Appendix C. Proofs for Infinite Families of Distributions

The proofs of the results on handling infinite Q sets follow analogously to the original proofs for finite $|Q|$, but with a few modifications to make use of results from the learning theory literature on infinite function classes. For completeness, we include the full details of these proofs here.

C.1. Proof of Theorem 12

We begin with the proof of Theorem 12. As above, we consider two key lemmas.

Lemma 14 *For P, R, and T as in Theorem 12, the distribution returns by Algorithm 1 satisfies* $\text{INV}(\hat{q}) \leq \varepsilon_2$ *with probability at least* $7/8$.

Proof Following the original proof above, let Invalid $= \{x : \text{INV}(x) = 1\}$ be the set of invalid points. Consider q^i for some i and any distribution $q \in Q$. If $q^i(\text{supp}(q) \cap \text{Invalid}) \geq \frac{\varepsilon_2}{R}$, then with probability at least $\frac{\varepsilon_2}{R}$ a sample generated from q^i lies in $\text{supp}(q) \cap \text{Invalid}$. Furthermore, we note that the VC dimension of the collection of sets $\{\text{supp}(q) \cap \text{Invalid} : q \in Q\}$ is at most d. Thus, with $T = \Theta(\frac{Rd}{\varepsilon_2}\log\frac{1}{\varepsilon_2})$ samples from q^i, the classic sample complexity result from PAC learning Vapnik and Chervonenkis (1974); Blumer et al. (1989) implies that with probability at least $1 - \frac{1}{8R}$, every $q \in Q$ with $q^i(\text{supp}(q) \cap \text{Invalid}) \geq \frac{\varepsilon_2}{R}$ has at least one of the T samples in $\text{supp}(q) \cap \text{Invalid}$. By a union bound, this holds for all i in the algorithm. Suppose this event holds.

In particular, this implies that if the algorithm returns in Step 9, so that the returned distribution $\hat{q} = q^i$ for some i, then $\text{INV}(q^i) = q^i(\text{supp}(q^i) \cap \text{Invalid}) < \frac{\varepsilon_2}{R} \leq \varepsilon_2$ as required. Furthermore, if the algorithm returns in Step 16 instead, then the above event implies that for every i, j with $i < j$, $q^i(\text{supp}(q^i) \cap \text{Invalid}) < \frac{\varepsilon_2}{R}$. Therefore, if we fix the value of i selected in Step 14, we have that

$$\begin{aligned}\text{INV}(\hat{q}) &= \mathbf{E}_{x\sim\hat{q}}\left[\text{INV}(x)\right] \\ &= \mathbf{E}_{x\sim q^i}\left[\text{INV}(x)\cdot\mathbb{I}\left[\exists j > i : x \in \text{supp}(q^j)\right]\right] \\ &\leq \sum_{j=i+1}^{R}\mathbf{E}_{x\sim q^i}\left[\text{INV}(x)\cdot\mathbb{I}\left[x \in \text{supp}(q^j)\right]\right] \\ &= \sum_{j=i+1}^{R} q^i(\text{supp}(q^j) \cap \text{Invalid}) \leq \sum_{j=i+1}^{R}\frac{\varepsilon_2}{R} < \varepsilon_2.\end{aligned}$$

■

Lemma 15 *For P, R, and T as in Theorem 12, the distribution $\hat{q}$ returned by Algorithm 1 satisfies $Loss(\hat{q}) \leq Loss(q^*) + \varepsilon_1$ with probability at least $7/8$.*

Proof Combining Corollary 2 of Haussler (1992) with Theorem 1 of Mendelson and Vershynin (2003), we conclude that for $P = \Theta\left(\frac{s(c\varepsilon_1/M)M^2}{\varepsilon_1^2}\log\frac{M}{\varepsilon_1}\right)$ samples from p, we have that the empirical loss $\overline{\text{Loss}}(q) \in \text{Loss}(q) \pm \frac{\varepsilon_1}{4}$ simultaneously for all $q \in Q$ with probability at least $15/16$. From here on, let us suppose this event occurs.

In that case, it must be that $\overline{\text{Loss}}(q^i) \leq \overline{\text{Loss}}(q^*)$. This is because the algorithm terminates if ever $q^i = q^*$ since q^* generates no invalid samples, and yet no q^i with $\overline{\text{Loss}}(q^i) > \overline{\text{Loss}}(q^*)$ will be considered before examining q^*.

This implies that at any point, we have that $\text{Loss}(q^i) \leq \overline{\text{Loss}}(q^i) + \frac{\varepsilon_1}{4} \leq \overline{\text{Loss}}(q^*) + \frac{\varepsilon_1}{4} \leq \text{Loss}(q^*) + \frac{\varepsilon_1}{2}$.

Therefore, in the case that the distribution that is output is $\hat{q} = q^i$ it will satisfy the given condition. To complete the proof we show the required property when returned distribution $\hat{q}$ is the improper meta-distribution.

In that case, we have that:

$$\begin{aligned}\text{Loss}(\hat{q}) &\leq \mathbf{E}_{x\sim p}\left[L\left(q_x^i \cdot \mathbb{I}\left[\exists j > i : x \in \text{supp}(q^j)\right]\right)\right] \\ &\leq \text{Loss}(q^i) + M \cdot \Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \\ &\leq \text{Loss}(q^*) + \frac{\varepsilon_1}{2} + M \cdot \Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]\end{aligned}$$

However, since a random index $i \sim \text{Uniform}(\{1, ..., R\})$ is chosen, we have that in expectation over this random choice

$$\begin{aligned}&\mathbf{E}_i\left[\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]\right] \\ &= \frac{1}{R}\sum_{i=1}^{R}\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \\ &= \frac{1}{R}\mathbf{E}_{x\sim p}\left[\sum_{i=1}^{R}\mathbb{I}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right]\right] \leq \frac{1}{R}\end{aligned}$$

where the last inequality follows since $\sum_{i=1}^{R}\mathbb{I}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \leq 1$ as only the largest i with $x \in \text{supp}(q^i)$ has that for all $j > i$, $x \notin \text{supp}(q^j)$.

By Markov's inequality, we have that with probability at least $15/16$, a random i will have

$$\Pr_{x\sim p}\left[x \in \text{supp}(q^i) \wedge \forall j > i : x \notin \text{supp}(q^j)\right] \leq \frac{16}{R}.$$

Therefore, the choice of $R = 32\frac{M}{\varepsilon_1} = \Theta\left(\frac{M}{\varepsilon_1}\right)$ guarantees that $\text{Loss}(\hat{q}) \leq \text{Loss}(q^*) + \varepsilon_1$. The overall failure probability is at most $1/16 + 1/16 = 1/8$. ■

Proof [Proof of Theorem 12] Theorem 12 follows immediately from the above two lemmas by a union bound. ■

C.2. Proof of Theorem 13

Next, the proof of Theorem 13 follows similarly to the original proof of Theorem 7, with a few important adjustments. As in the statement of the theorem, we consider running Algorithm 2′ with parameters ε_1, ε_2, and $\alpha + \varepsilon_2$. As in the proof of Theorem 7, we proceed by establishing three key lemmas. As much of this proof essentially follows by *plugging in* the altered set D (from the new Step 4) to the arguments of the original proofs above, in the proofs of these lemmas we only highlight the reasons for which this substitution remains valid and yields the stated result.

Lemma 16 *With probability at least* $14/15$*, the loop at Line 7 of Algorithm 2′ requires at most* $O\left(\frac{\mathrm{fat}_{c\varepsilon_2}(Q)}{\varepsilon_2}\log\left(\frac{1}{\varepsilon_2}\right)\right)$ *iterations for each* ℓ*.*

Proof We invoke the original argument from the proof of Lemma 9 verbatim, except that rather than bounding the initial size $|D|$ in Step 4 by $|Q|$, we use the fact that Step 4 in Algorithm 2′ initializes $|D|$ to the minimal size of an ε_2-cover of $\{q \in Q | \overline{\mathrm{Loss}}(q) \le \ell\}$, which is at most the size of a minimal ε_2-cover of Q (under the $L_1(\mu_0)$ pseudo-metric). Thus, Theorem 1 of Mendelson and Vershynin (2003) implies that, for every ℓ, this initial set D satisfies

$$\log(|D|) = O\left(\mathrm{fat}_{c\varepsilon_2}(Q)\log\left(\frac{1}{\varepsilon_2}\right)\right). \tag{3}$$

The lemma then follows from the same argument as in the proof of Lemma 9. ■

Lemma 17 *With probability at least* $14/15$*, if at any step a distribution* μ'_D *is output,* $\mathrm{INV}(\mu'_D) \le \alpha + 2\varepsilon_2$*.*

Proof The argument remains identical to the proof of Lemma 10, except again substituting for $\log|Q|$ the quantity on the right hand side of (3), and substituting $\alpha + \varepsilon_2$ for α. ■

Lemma 18 *With probability at least* $14/15$*, if at any step a distribution* μ'_D *is output, Loss*$(\mu'_D) \le \ell + 2\varepsilon_1/3$*, where* ℓ *is the step at which the distribution was output.*

Proof Combining Corollary 2 of Haussler (1992) with Theorem 1 of Mendelson and Vershynin (2003) implies that the choice $n_1 = \Theta\left(\frac{s(c\varepsilon_1/M)M^2}{\varepsilon_1^2}\log\left(\frac{M}{\varepsilon_1}\right)\right)$ suffices to guarantee every $q \in Q$ has $\overline{\mathrm{Loss}}(q)$ within $\pm\varepsilon_1/3$ of $\mathrm{Loss}(q)$. Substituting this argument for the final step in the proof of Lemma 11, and leaving the rest of that proof intact, this result follows. ■

Proof [Proof of Theorem 13] The proof of Theorem 13 concludes by observing that, upon reaching ℓ within $\varepsilon_1/3$ of $\mathrm{Loss}(q^*)$ (where q^* is the optimal distribution), the closest (in $L_1(\mu_0)$) element q of the corresponding D set will have $\mathrm{INV}(q) \le \mathrm{INV}(q^*) + \varepsilon_2 \le \alpha + \varepsilon_2$, and (by definition of D) $\mathrm{Loss}(q) \le \mathrm{Loss}(q^*) + \varepsilon_1/3$. Thus, this q will never be eliminated (assuming all estimates involving its loss and validity are accurate, which happens with probability at least $19/20$). Combining this with Lemma 18, if we output $\hat{q}$, then $\mathrm{Loss}(\hat{q}) \le \mathrm{Loss}(q^*) + \varepsilon_1$. ■