Efficient Algorithms for Learning from Coarse Labels

Dimitris Fotakis FOTAKIS@CS.NTUA.GR
National Technical University of Athens

Alkis Kalavasis KALAVASISALKIS@MAIL.NTUA.GR
National Technical University of Athens

Vasilis Kontonis KONTONIS@WISC.EDU
University of Wisconsin-Madison

Christos Tzamos TZAMOS@WISC.EDU
University of Wisconsin-Madison

Editors: Mikhail Belkin and Samory Kpotufe

Abstract

For many learning problems one may not have access to fine grained label information; e.g., an image can be labeled as husky, dog, or even animal depending on the expertise of the annotator. In this work, we formalize these settings and study the problem of learning from such coarse data. Instead of observing the actual labels from a set $\mathcal{Z}$, we observe coarse labels corresponding to a partition of $\mathcal{Z}$ (or a mixture of partitions).

Our main algorithmic result is that essentially any problem learnable from fine grained labels can also be learned efficiently when the coarse data are sufficiently informative. We obtain our result through a generic reduction for answering Statistical Queries (SQ) over fine grained labels given only coarse labels. The number of coarse labels required depends polynomially on the information distortion due to coarsening and the number of fine labels $|\mathcal{Z}|$.

We also investigate the case of (infinitely many) real valued labels focusing on a central problem in censored and truncated statistics: Gaussian mean estimation from coarse data. We provide an efficient algorithm when the sets in the partition are convex and establish that the problem is NP-hard even for very simple non-convex sets. [1]

Keywords: Coarse Labels, Statistical Queries, Censored Statistics

1. Introduction

Supervised learning from labeled examples is a classical problem in machine learning and statistics: given labeled examples, the goal is to train some model to achieve low classification error. In most modern applications, where we train complicated models such as neural nets, large amounts of labeled examples are required. Large datasets such as Imagenet, Russakovsky et al. (2015), often contain thousands of different categories such as animals, vehicles, etc., each one of those containing many *fine grained* subcategories: animals may contain dogs and cats and dogs may be further split into different breeds etc. In the last few years, there have been many works that focus on fine grained recognition, Guo et al. (2018); Chen et al. (2018); Touvron et al. (2020); Qin et al. (2020); Lei et al. (2017); Jiao et al. (2019, 2020); Bukchin et al. (2020); Taherkhani et al. (2019). Collecting a sufficient amount of accurately labeled training examples is a hard and expensive task that often requires hiring experts to annotate the examples. This has motivated the problem of

1. The full version is available on arXiv with the same title and contains the proofs of all results discussed in this paper.

learning from *coarsely* labeled datasets, where a dataset is not fully annotated with fine grained labels but a combination of fine, e.g., cat, and coarse labels, e.g., animal, is given, Deng et al. (2013); Ristin et al. (2015).

Inference from coarse data naturally arises also in unsupervised, i.e., distribution learning settings: instead of directly observing samples from the target distribution, we observe "representative" points that correspond to larger sets of samples. For example, instead of observing samples from a real valued random variable, we round them to the closest integer. An important unsupervised problem that fits in the coarse data framework is censored statistics, Cohen (2016); Wolynetz (1979); Breen et al. (1996); Schneider (1986). Interval censoring, that arises in insurance adjustment applications, corresponds to observing points in some interval and point masses at the endpoints of the interval instead of observing fine grained data from the whole real line. Moreover, the problem of learning the distribution of the output of neural networks with non-smooth activations (e.g., ReLU networks, Wu et al. (2019)) also fits in our model of distribution learning with coarse data, see Figure 1(c).

Even though the problem of learning from coarsely labeled data has attracted significant attention from the applied community, from a theoretical perspective little is known. In this work, we provide efficient algorithms that work in both the supervised and the unsupervised coarse data settings.

1.1. Our Model and Results

We start by describing the generative model of coarsely labeled data in the supervised setting. We model coarse labels as subsets of the domain of all possible fine labels. For example, assume we hire an expert on dog breeds and an expert on cat breeds to annotate a dataset containing images of dogs and cats. With probability $1/2$, we get samples labeled by the dog expert, i.e., labeled according to the partition $\{\text{cat} = \{\text{persian cat}, \text{bengal cat}, \dots\}, \{\text{maltese dog}\}, \{\text{husky dog}\}, \dots\}$. On the other hand, the cat expert will provide a fine grained partition over cat breeds and will group together all dog breeds. Our coarse data model captures exactly this mixture of different label partitions.

Definition 1 (Generative Process of Coarse Data with Context) *Let $\mathcal{X}$ be an arbitrary domain, and let $\mathcal{Z} = \{1, \dots, k\}$ be the discrete domain of all possible fine labels. We generate coarsely labeled examples as follows:*

1. *Draw a finely labeled example (x, z) from a distribution $\mathcal{D}$ on $\mathcal{X} \times \mathcal{Z}$.*
2. *Draw a coarsening partition $\mathcal{S}$ (of $\mathcal{Z}$) from a distribution π.*
3. *Find the unique set $S \in \mathcal{S}$ that contains the fine label z.*
4. *Observe the coarsely labeled example (x, S).*

We denote $\mathcal{D}_\pi$ the distribution of the coarsely labeled example (x, S).

In the supervised setting, our main focus is to answer the following question.

Question 2 *Can we train a model, using coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, that classifies finely labeled examples $(x, z) \sim \mathcal{D}$ with accuracy comparable to that of a classifier that was trained on examples with fine grained labels?*

Definition 1 does not impose any restrictions on the distribution over partitions π. It is clear that if partitions are very rough, e.g., we split $\mathcal{Z}$ into two large disjoint subsets, we lose information about the fine labels and we cannot hope to train a classifier that performs well over finely labeled examples. In order for Question 2 to be information theoretically possible, we need to assume that the partition distribution π preserves fine-label information. The following definition quantifies this by stating that reasonable partition distributions π are those that preserve the total variation distance between different distributions supported on the domain of the fine labels $\mathcal{Z}$. We remark that the following definition does not require $\mathcal{D}$ to be supported on pairs (x, S) but is a general statement for the unsupervised version of the problem, see also Definition 9.

Definition 3 (Information Preserving Partition Distribution) *Let $\mathcal{Z}$ be any domain and let $\alpha \in (0, 1]$. We say that π is an α-information preserving partition distribution if for every two distributions $\mathcal{D}^1, \mathcal{D}^2$ supported on $\mathcal{Z}$, it holds that $\mathrm{TV}(\mathcal{D}^1_\pi, \mathcal{D}^2_\pi) \geq \alpha\ \mathrm{TV}(\mathcal{D}^1, \mathcal{D}^2)$, where $\mathrm{TV}(\mathcal{D}^1, \mathcal{D}^2)$ is the total variation distance of $\mathcal{D}^1$ and $\mathcal{D}^2$.*

For example, the partition distribution defined in the dog/cat dataset scenario, discussed before Definition 1, is $1/2$-information preserving, since we observe fine labels with probability $1/2$. In this case, it is easy, at the expense of losing the statistical power of the coarse labels, to combine the finely labeled examples from both experts in order to obtain a dataset consisting only of fine labels. However, our model allows the partitions to have arbitrarily complex combinatorial structure that makes the process of "inverting" the partition transformation computationally challenging. For example, specific fine labels may be complicated functions of coarse labels: "medium sized" and "pointy ears" and "blue eyes" may be mapped to the "husky dog" fine label.

Our first result is a positive answer to Question 2 in essentially full generality: we show that concept classes that are efficiently learnable in the Statistical Query (SQ) model, Kearns (1998), are also learnable from coarsely labeled examples. Our result is similar in spirit with the result of Kearns (1998), where it is proved that SQ learnability implies learnability under random classification noise.

Informal Theorem 1 (SQ Learnability implies Learnability from Coarse Examples) *Any concept class $\mathcal{C}$ that is efficiently learnable with M statistical queries from finely labeled examples $(x, z) \sim \mathcal{D}$, can be efficiently learned from $O(\mathrm{poly}(k/\alpha)) \cdot M$ coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$ under any α-information preserving partition distribution π.*

Statistical Queries are queries of the form $\mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, z)]$ for some query function $q(x, z)$. It is known that almost all known machine learning algorithms Aslam and Decatur (1998); Blum et al. (1998, 2005); Dunagan and Vempala (2008); Balcan and Feldman (2015); Feldman et al. (2017a) can be implemented in the SQ model. In particular, in Feldman et al. (2017b), it is shown that (Stochastic) Gradient Descent can be simulated by statistical queries. This implies that our result can be applied, even in cases where it is not possible to obtain formal optimality guarantees, e.g., training deep neural nets. We can train such models using coarsely labeled data and guarantee the same performance as if we had direct access to fine labels. As another application, we consider the problem of multiclass logistic regression with coarse labels. It is known, see e.g., Friedman et al. (2001), that given finely labeled examples $(x, z) \sim \mathcal{D}$, the likelihood objective for multiclass logistic regression is concave with respect to the weight matrix. Even though the likelihood objective is no-longer concave when we consider coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$, our theorem bypasses this difficulty and allows us to efficiently perform multiclass logistic regression with coarse labels.

Formally, we design an algorithm (Algorithm 1) that, given coarsely labeled examples (x, S), efficiently simulates statistical queries over finely labeled examples (x, z). Surprisingly, the runtime and sample complexity of our algorithm do not depend on the combinatorial structure of the partitions, but only on the number of fine labels k and the information preserving parameter α of the partition distribution π.

Theorem 4 (SQ from Coarsely Labeled Examples) *Consider a distribution $\mathcal{D}_\pi$ over coarsely labeled examples in $\mathbb{R}^d \times [k]$, (see Definition 1) with α-information preserving partition distribution π. Let $q : \mathbb{R}^d \times \mathcal{Z} \to [-1, 1]$ be a query function, that can be evaluated on any input in time T, and $\tau, \delta \in (0, 1)$. There exists an algorithm (Algorithm 1), that draws $N = \widetilde{O}(k^4/(\tau^3\alpha^2)\log(1/\delta))$ coarsely labeled examples from $\mathcal{D}_\pi$ and, in $\mathrm{poly}(N, T)$ time, computes an estimate $\hat{r}$ such that, with probability at least $1 - \delta$, it holds $\left|\mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, z)] - \hat{r}\right| \leq \tau$.*

Learning Parametric Distributions from Coarse Samples. In many important applications, instead of a discrete distribution over fine labels, a continuous parametric model is used. A popular example is when the domain $\mathcal{Z}$ of Definition 1 is the entire Euclidean space $\mathbb{R}^d$, and the distribution of finely labeled examples is a Gaussian distribution whose parameters possibly depend on the context x. Such censored regression settings are known as Tobit models Tobin (1958); Maddala (1986); Gourieroux (2000). Lately, significant progress has been made from a computational point of view in such censored/truncated settings in the distribution specific setting, e.g., when the underlying distribution is Gaussian Daskalakis et al. (2018); Kontonis et al. (2019), mixtures of Gaussians Nagarajan and Panageas (2019), linear regression Daskalakis et al. (2019); Ilyas et al. (2020); Daskalakis et al. (2020). In this distribution specific setting, we consider the most fundamental problem of learning the mean of a Gaussian distribution given coarse data.

Definition 5 (Coarse Gaussian Data) *Consider the Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}^\star)$, with mean $\boldsymbol{\mu}^\star \in \mathbb{R}^d$ and identity covariance matrix. We generate a sample as follows:*

1. *Draw $\boldsymbol{z}$ from $\mathcal{N}(\boldsymbol{\mu}^\star)$.*
2. *Draw a partition $\mathcal{S}$ (of $\mathbb{R}^d$) from π.*
3. *Observe the set $S \in \mathcal{S}$ that contains $\boldsymbol{z}$.*

We denote the distribution of S as $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$.

We first study the above problem, from a computational viewpoint. For the corresponding problems in censored and truncated statistics no geometric assumptions are required for the sets: in Daskalakis et al. (2018) is was shown that an efficient algorithm exists for arbitrarily complex truncation sets. In contrast in our more general model of coarse data we show that having sets with geometric structure is necessary. In particular we require that every set of the partition is convex, see Figure 1(b,c). We show that when the convexity assumption is dropped, learning from coarse data is a computationally hard problem even under a mixture of very simple sets.

Theorem 6 (Hardness of Matching the Observed Distribution with General Partitions) *Let π be a general partition distribution. Unless $\mathrm{P} = \mathrm{NP}$, no algorithm with sample access to $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, can compute, in $\mathrm{poly}(d)$ time, a $\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d$ such that $\mathrm{TV}(\mathcal{N}_\pi(\widetilde{\boldsymbol{\mu}}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star)) < 1/d^c$ for some absolute constant $c > 1$.*

We prove our hardness result using a reduction from the well known MAX-CUT problem, which is known to be NP-hard, even to approximate Håstad (2001). In our reduction, we use partitions that consist of simple sets: fat hyperplanes, ellipsoids and their complements: the computational hardness of this problem is rather inherent and not due to overly complicated sets.

On the positive side, we identify a geometric property that enables us to design a computationally efficient algorithm for this problem: namely we require all the sets of the partitions to be *convex*, e.g., Figure 1(b,c). We remark that having finite or countable subsets, is not a requirement of our model. For example, we can handle convex partitions of the form (c) that correspond to the output distribution of a ReLU neural network, see Wu et al. (2019). We continue with our theorem for learning Gaussians from coarse data.

Theorem 7 (Gaussian Mean Estimation with Convex Partitions) *Let $\epsilon, \delta \in (0,1)$. Consider the generative process of coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$. Assume that the partition distribution π is α-information preserving and is supported on convex partitions of $\mathbb{R}^d$. There exists an algorithm, that draws $\widetilde{O}(d/(\epsilon^2\alpha^2)\log(1/\delta))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, runs in time polynomial in the number of samples, and computes an estimate $\widetilde{\boldsymbol{\mu}}$ that satisfies $\mathrm{TV}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$ with probability at least $1-\delta$.*

Our algorithm for mean estimation of a Gaussian distribution relies on the likelihood being concave when the partitions are convex. We remark that, similar to our approach, one can use the concavity of likelihood to get efficient algorithms for regression settings, e.g., Tobit models, where the mean of the Gaussian is given by a linear function of the context $\boldsymbol{A}\boldsymbol{x}$ for some unknown matrix $\boldsymbol{A}$.

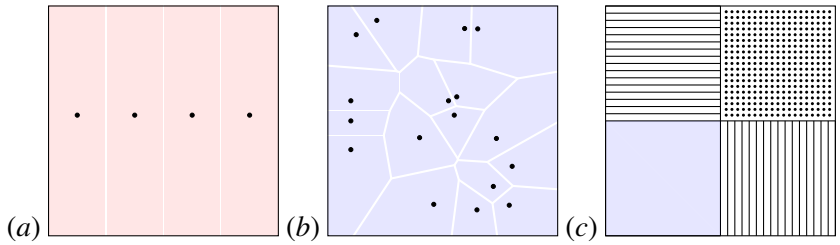

Figure 1: (a) is a very rough partition, that makes learning the mean impossible: Gaussians $\mathcal{N}((0,z))$ centered along the same vertical line $(0,z)$ assign exactly the same probability to all cells of the partitions and therefore, $\mathrm{TV}(\mathcal{N}_\pi((0,z_1)), \mathcal{N}_\pi((0,z_2))) = 0$: it is impossible to learn the second coordinate of the mean. (b) is a convex partition of $\mathbb{R}^2$, that makes recovering the Gaussian possible. (c) is the convex partition corresponding to the output distribution of one layer ReLU networks. When both coordinates are positive, we observe a fine sample (black points correspond to singleton sets). When exactly one coordinate (say $\boldsymbol{x}_1$) is positive, we observe the line $L_z = \{\boldsymbol{x} : \boldsymbol{x}_2 < 0, \boldsymbol{x}_1 = z > 0\}$ that corresponds to the ReLU output $(\boldsymbol{x}_1, 0)$. If both coordinates are negative, we observe the set $\{\boldsymbol{x} : \boldsymbol{x}_1 < 0, \boldsymbol{x}_2 < 0\}$, that corresponds to the point $(0,0)$.

1.2. Related Work

Our work is closely related to the literature of learning from censored-truncated data and learning with noise. There has been a large number of recent works dealing inference with truncated data from a Gaussian distribution Daskalakis et al. (2018); Kontonis et al. (2019), mixtures of Gaussians Nagarajan and Panageas (2019), linear regression Daskalakis et al. (2019); Ilyas et al. (2020);

Daskalakis et al. (2020), sparse Graphical models Bhattacharyya et al. (2021) or Boolean product distributions Fotakis et al. (2020). A significant feature of our work is that it can capture the closely related field of censored statistics Cohen (2016); Breen et al. (1996); Wolynetz (1979).

The area of robust statistics Huber (2004) is also very related to our work as it also deals with biased data-sets and aims to identify the distribution that generated the data. Recently, there has been a large volume of theoretical work for computationally-efficient robust estimation of high-dimensional distributions Diakonikolas et al. (2016); Charikar et al. (2017); Lai et al. (2016); Diakonikolas et al. (2017a, 2018); Klivans et al. (2018); Hopkins and Li (2019); Diakonikolas et al. (2019); Cheng et al. (2020); Bakshi et al. (2020) in the presence of arbitrary corruptions to a small ε fraction of the samples.

The line of research dealing with statistical queries Kearns (1998); Blum et al. (1998); Feldman et al. (2015, 2017b); Feldman (2017); Feldman et al. (2017a); Diakonikolas et al. (2017b, 2020b) is closely related to one of our main results (Theorem 4). It is generally believed that SQ algorithms capture all reasonable machine learning algorithms Aslam and Decatur (1998); Blum et al. (1998, 2005); Dunagan and Vempala (2008); Feldman et al. (2017a); Balcan and Feldman (2015); Feldman et al. (2017b) and there is a rich line of research indicating SQ lower-bounds for these classes of algorithms Feldman et al. (2017a); Diakonikolas et al. (2017b); Shamir (2018); Vempala and Wilmes (2019); Diakonikolas et al. (2020b,a); Goel et al. (2020a,b).

2. Notation and Preliminaries

We let $[n] = \{1, \dots, n\}$. We use lowercase bold letters $\boldsymbol{x}$ to denote vectors and capital bold letters $\boldsymbol{X}$ for matrices. We let $\boldsymbol{x}_i$ be the i-th coordinate of $\boldsymbol{x}$. We let $\|\boldsymbol{x}\|_p$ denote the L_p norm of $\boldsymbol{x}$. The probability simplex is denoted by Δ^n and discrete distributions $\mathcal{D}$ supported on $[n]$ will usually be represented by their associated probability vectors $\boldsymbol{p} \in \Delta^n$. For any distribution $\mathcal{D}$, we overload the notation and we use the same notation for the corresponding density and denote $\mathcal{D}(S) = \sum_{x \in S} \mathcal{D}(x)$ for any $S \subseteq [n]$. The d-dimensional Gaussian distribution will be denoted by $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. When the covariance matrix is known, we simplify to $\mathcal{N}(\boldsymbol{\mu})$. We denote Φ (resp. ϕ) the cdf (resp. pdf) of the standard Normal distribution. The total variation distance of $\boldsymbol{p}, \boldsymbol{q} \in \Delta^n$ is $\mathrm{TV}(\boldsymbol{p}, \boldsymbol{q}) = \max_{S \subseteq [n]} \boldsymbol{p}(S) - \boldsymbol{q}(S) = \|\boldsymbol{p} - \boldsymbol{q}\|_1/2$. For a random variable x, we let $\mathbf{E}[x], \mathbf{V}(x), \mathbf{Cov}(x)$ be the expected value, the variance and the covariance of x. For a joint distribution $\mathcal{D}$ of two random variables x and z over the space $\mathcal{X} \times \mathcal{Z}$, we let $\mathcal{D}_x$ (resp. $\mathcal{D}_z$) be the marginal distribution of x (resp. z). Let $\mathcal{D}$ be a joint distribution over labeled examples $\mathcal{X} \times \mathcal{Z}$, with $\mathcal{X}$ be the input space and $\mathcal{Z}$ the label space. A statistical query (SQ) oracle $\mathrm{STAT}(\mathcal{D}, \tau)$ with tolerance parameter $\tau \in [0, 1]$ takes as input a statistical query defined by a real-valued function $q : \mathcal{X} \times \mathcal{Z} \to [-1, 1]$ and outputs an estimate of $\mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, z)]$ that is accurate to within an additive $\pm\tau$.

3. Supervised Learning from Coarse Data

In this section, we consider the problem of *supervised* learning from coarse data. In this setting, there exists some underlying distribution over finely labeled examples, $\mathcal{D}$. However, we have sample access only to the distribution associated with coarsely labeled examples $\mathcal{D}_\pi$, see Definition 1. As discussed in Section 1, under this setting, even problems that are naturally convex when we have access to examples with fine labels, become non-convex when we introduce coarse labels (e.g.,

multiclass logistic regression). The main result of this section is Theorem 4, which allows us to compute statistical queries over finely labeled examples.

3.1. Overview of the Proof of Theorem 4

In order to simulate a statistical query we take a two step approach. Our first building block considers the unsupervised version of the problem, see Definition 9, i.e., we marginalize the context x and try to learn the distribution of the fine labels z given coarse samples S. This can be viewed as learning a general discrete distribution supported on $\mathcal{Z} = \{1, \ldots, k\}$ given coarse samples, i.e., subsets of $\mathcal{Z}$. We show that, when the partition distribution π is α-information preserving, this can be done efficiently, see Proposition 10. Our algorithm (Algorithm 1) exploits the fact that even though in general having coarse data results in non-concave likelihood objectives, when we consider parametric models, this is not true when we maximize over all discrete distributions. In Proposition 10, we show that $\widetilde{O}(k/(\epsilon\alpha)^2)$ samples are sufficient for this step. For the details of this step, see Subsection 3.2.

Using the above algorithm, one could try to separately learn the marginal distribution over x, $\mathcal{D}_x$ and the distribution of the fine labels z *conditional on some fixed* x; let us denote this distribution as $\mathcal{D}_z^x$. Then one could generate finely labeled examples (x, z) and use them to estimate the query $\mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, z)]$. The reason that this naive approach fails is that it requires many coarse examples (x, S) with exactly the same value of x. Unless the domain $\mathcal{X}$ is very small, the probability that we observe samples with the same value of x is going to be tiny. In order to overcome this obstacle, at a high level, our approach is to split the domain $\mathcal{X}$ into larger sets and then, learn the conditional distribution of the labels, not on a fixed point x, but on these larger sets of non-trivial mass.

Intuitively, in order to have an effective partition of the domain $\mathcal{X}$, we want to group together points x whose values $q(x, z)$ are close. Since z belongs in a discrete domain $\mathcal{Z} = [k]$, we can decompose the query $q(x, z)$ as $q(x, z) = \sum_{i=1}^{k} q(x, i)\mathbf{1}\{z = i\}$. We estimate the value of $\mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, i)\mathbf{1}\{z = i\}]$ separately. To find a suitable reweighting of the domain $\mathcal{X}$, we perform rejection sampling, accepting a pair $(x, S) \sim \mathcal{D}$ with probability $q(x, i)$ [2]: points x that have small value $q(x, i)$ contribute less in the expectation and are less likely to be sampled. After we perform this rejection sampling process based on x, we have pairs (x, S), conditional that x was accepted. Now, using our previous maximum likelihood learner of Proposition 10 we learn the marginal distribution over fine labels and use it to answer the query. We provide the details of this rejection sampling step in the full proof of Theorem 4, see Subsection 3.3.

For a description of the corresponding algorithm that simulates statistical queries, see Algorithm 1. To keep the presentation simple we state the algorithm for the case where the query function $q(x, y)$ is positive. It is straightforward to generalize it for general queries, see Subsection 3.3.

Remark 8 (Empirical Likelihood Approach) *One could try to use the empirical likelihood directly over the coarsely labeled data (as defined in Owen (2001)). However, in general, these empirical likelihood objectives are non-convex when the data are coarse and therefore it is computationally hard to optimize them directly. Our approach for simulating statistical queries consists of two ingredients: reweighting the feature space via rejection sampling in order to group together points and learning discrete distributions from coarse data. To learn the discrete distributions (see Section 3.2), we use a (direct) empirical likelihood approach similar to that of Owen (1988); Owen*

2. It is easy to handle the case where this function takes negative values, see the proof of Theorem 4.

et al. (1990); Owen (2001). Our main contribution is the use of rejection sampling to reduce the initial non-convex problem to the special case of learning a discrete distribution (with small support) from coarse data which, as we prove, is a tractable (convex) problem. For more connections with censored statistics techniques, we refer the reader to Thomas and Grunkemeier (1975); Owen (1988); Gill et al. (1997); Owen (2001).

Algorithm 1 Statistical Queries from Coarse Labels.

Input: Query $q : \mathcal{X} \times \mathcal{Z} \mapsto (0, 1]$, accuracy $\tau \in [0, 1]$, confidence $\delta \in [0, 1]$.
Oracle: Access to coarsely labeled samples $(x, S) \sim \mathcal{D}_\pi$.
Output: Estimate $\hat{r}$ such that $\left| \mathbf{E}_{(x,z)\sim\mathcal{D}}[q(x, z)] - \hat{r} \right| \leq \tau$ with probability at least $1 - \delta$.
procedure STATQUERY(q, τ, δ)
    Compute $\hat{r}_i \leftarrow \mathrm{SQ}(q, i, O(\tau/k), \delta)$.
    Output $\hat{r} \leftarrow \sum_{i=1}^{k} \hat{r}_i$.
procedure SQ(q, i, ρ, δ)
    Draw $N_1 = \widetilde{\Theta}\big(\frac{\log(1/\delta)}{\rho^2}\big)$ samples (x_j, S_j) from $\mathcal{D}_\pi$.
    Compute $\widehat{\mu}_i \leftarrow \frac{1}{N_1} \sum_{j=1}^{N_1} q(x_j, i)$.
    if $\widehat{\mu}_i \leq \rho$ **do**
        Output $\widehat{r}_i \leftarrow 0$.
    end
    Draw $N_2 = \widetilde{\Theta}\big(\frac{k \log(1/\delta)}{\rho^3 \alpha^2}\big)$ samples (x_j, S_j) from $\mathcal{D}_\pi$. ▷ $\widetilde{\Theta}\big(\frac{k^4 \log(1/\delta)}{\tau^3 \alpha^2}\big)$ *examples overall.*
    $T_{accept} \leftarrow \emptyset$. ▷ *Training set of accepted samples.*
    Add S_j in T_{accept} with probability $q(x_j, i)$, $\forall j \in [N_2]$. ▷ *Rejection Sampling Process.*
    Compute $\widetilde{\mathcal{D}}$ using Proposition 10 with input $(T_{accept}, \rho, \delta)$.
    Output $\hat{r}_i \leftarrow \widetilde{\mu}_i \, \widetilde{\mathcal{D}}(i)$.

3.2. Learning Marginals Over Fine Labels

In this subsection, we deal with *unsupervised* learning from coarse data in discrete domains. Although this is an ingredient of our main result for simulating statistical queries in a supervised setting where labeled data (x, S) are given, the result of this section does not depend on the points x and concerns the unsupervised version of the problem. To keep the notation simple, we will use $\mathcal{D}$ to denote a distribution over finite labels $\mathcal{Z}$.

Definition 9 (Generative Process of Coarse Data) *Let $\mathcal{Z}$ be a discrete domain and $\mathcal{D}$ be a distribution supported on $\mathcal{Z}$. Moreover, let π be a distribution supported on partitions of $\mathcal{Z}$. We consider the following generative process:*

1. *Draw z from $\mathcal{D}$.*
2. *Draw a partition $\mathcal{S}$ from the distribution over all partitions π.*
3. *Observe the set $S \in \mathcal{S}$ that contains z.*

We denote the distribution of S as $\mathcal{D}_\pi$.

The assumption that we require is that the partition distirbution π is α-information preserving, see Definition 3. At this point we give some examples of information preserving partition distributions. We first observe that $\alpha = 0$ if and only if the problem is not identifiable. For instance, if π is supported only on the partition $\mathcal{S} = \{\{1, 2\}, \{3, \ldots, k\}\}$, the problem is not identifiable, since, for example, the fine label 1 is indistinguishable from the fine label 2. The value $\alpha = 1$ is attained when the partition totally preserves the distribution distance. Intuitively, the value $1 - \alpha$ corresponds to the distortion that the coarse labeling introduces to a fine-labeled dataset.

In many cases most fine labels may be missing. Consider two data providers that use different methods to round their samples. The rounding's uncertainty can be viewed as a coarse labeling of the data. Assume that we add discrete (balanced Bernoulli) noise ξ to some true value $x \in [0..k]$. Consider two partitions $\{\mathcal{S}_1, \mathcal{S}_2\}$ with $\mathcal{S}_1 = \{\{0, 1\}, \{2, 3\}, \ldots, \{k - 1, k\}, \{k + 1\}\}$ and $\mathcal{S}_2 = \{\{0\}, \{1, 2\}, \ldots \{k - 1, k\}\}$. Observe that, when $x + \xi$ is odd, we can think of the rounded sample, as a draw from $\mathcal{S}_1$ and when $x + \xi$ is even, as a draw from $\mathcal{S}_2$. This example shows that we can capture the problem of deconvolution of two distributions $\mathcal{D}_1, \mathcal{D}_2$, where one of them is known and we observe samples $x_1 + x_2, x_i \sim \mathcal{D}_i$.

The following proposition establishes the sample complexity of unsupervised learning of discrete distributions with coarse data. Our goal is to compute an estimate of the discrete distribution $\mathcal{D}^\star$ with probability vector $\boldsymbol{p}^\star \in \Delta^k$ from N coarse samples $S_1, \ldots, S_N$ drawn from the distribution $\mathcal{D}^\star_\pi$. Our algorithm maximizes the empirical likelihood. Analyzing the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{p}) = \frac{1}{N} \sum_{n=1}^N \log\big(\sum_{i \in S_n} \boldsymbol{p}_i\big)$, where $\boldsymbol{p} \in \Delta^k$ is a guess probability vector, we observe that the problem is concave and, therefore, can be efficiently optimized (e.g., by gradient descent).

Proposition 10 *Let $\mathcal{Z}$ be a discrete domain of cardinality k and let $\mathcal{D}$ be a distribution supported on $\mathcal{Z}$. Moreover, let π be an α-information preserving partition distribution for some $\alpha \in (0, 1]$. Then, with $N = \widetilde{O}(k/(\epsilon^2\alpha^2)\log(1/\delta))$ samples from $\mathcal{D}_\pi$ and in time polynomial in the number of samples N, we can compute a distribution $\widetilde{\mathcal{D}}$ supported on $\mathcal{Z}$ such that $\mathrm{TV}(\widetilde{\mathcal{D}}, \mathcal{D}) \leq \epsilon$.*

3.3. The Proof of Theorem 4

In this subsection, we prove Theorem 4. Recall that we have sample access only to coarsely labeled examples $(x, S) \sim \mathcal{D}_\pi$. The key idea is to perform rejection sampling on each coarse sample (x, S) with acceptance probability $q(x, j)$ for any fine label $j \in \mathcal{Z}$. Because of the rejection sampling process, this marginal distribution is not the marginal of $\mathcal{D}$ on the fine labels $\mathcal{Z}$, but the marginal of $\mathcal{D}$ on the fine labels, conditional on the accepted samples. However, the task of estimating from this marginal distribution can be still reduced to the unsupervised problem, see Proposition 10 of the previous section. Consider an arbitrary query function $q : \mathcal{X} \times \mathcal{Z} \to [-1, 1]$ and, without loss of generality, let $\mathcal{Z} = [k]$. Recall that $\mathcal{D}$ is the joint probability distribution on the finely labeled examples (x, z). We have that

$$\mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}[q(x, z)] = \sum_{j=1}^k \mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}\Big[q(x, j)\mathbf{1}\{z = j\}\Big] = \sum_{j=1}^k \mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}\Big[q_j(x)\mathbf{1}\{z = j\}\Big]. \tag{1}$$

Since we would like to estimate the expectation of the query $q(x, z)$ with tolerance τ, it suffices to estimate the expectation of each query $q_j(x)\mathbf{1}\{z = j\}$ with tolerance τ/k for any $j \in [k]$. Hence,

it suffices to estimate expectations of the form $\mathbf{E}_{(x,z)\sim\mathcal{D}}[f(x)\mathbf{1}\{z = j\}]$ for arbitrary functions $f : \mathcal{X} \to [0,1]$[3] and $j \in [k]$.

Let $\mathcal{D}_x$ denote the marginal distribution of the examples $x \in \mathcal{X}$. The algorithm performs rejection sampling. Each coarsely labeled example $(x, S) \sim \mathcal{D}_\pi$ is accepted with probability $f(x)$, that does not depend on the coarse label S. Hence, the rejection sampling process induces a distribution $\mathcal{D}^f$ over finely labeled examples $(x, z) \in \mathcal{X} \times \mathcal{Z}$ with density

$$\mathcal{D}^f(x,z) = \frac{f(x)}{\mathbf{E}_{x\sim\mathcal{D}_x}[f(x)]}\mathcal{D}(x,z)\,.$$

We remark that, we do not have sample access to $\mathcal{D}^f$ because we do not have sample access to the distribution $\mathcal{D}$ of the fine examples; we introduced the above notation for the purposes of the proof. Similarly, to $\mathcal{D}_x$, we define $\mathcal{D}_x^f$ to be the marginal distribution of x conditional on its acceptance, i.e.,

$$\mathcal{D}_x^f(x) = \frac{f(x)}{\mathbf{E}_{x\sim\mathcal{D}_x}[f(x)]}\mathcal{D}_x(x)\,. \tag{2}$$

Let $\mathcal{D}_z$ denote the marginal distribution of the fine labels $[k]$ and let $\mathcal{D}_z(\cdot|x)$ be the marginal distribution conditional on the example x. We have that

$$\mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}\Big[f(x)\mathbf{1}\{z = j\}\Big] = \int_{\mathcal{X}} f(x)\mathcal{D}(x,j)dx = \int_{\mathcal{X}} f(x)\mathcal{D}_x(x)\mathcal{D}_z(j|x)dx\,.$$

The above expectation can be equivalently written, by multiplying and dividing by $\mathcal{D}_x^f$,

$$\mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}\Big[f(x)\mathbf{1}\{z = j\}\Big] = \int_{\mathcal{X}} \Big(\frac{f(x)\mathcal{D}_x(x)}{\mathcal{D}_x^f(x)}\Big)\Big(\mathcal{D}_x^f(x)\mathcal{D}_z(j|x)\Big)dx\,.$$

The first term in the integral is equal to $\mathbf{E}_{x\sim\mathcal{D}_x}[f(x)]$, by substituting Equation (2) and, hence, is constant. The second term corresponds to the probability of observing the fine label j, given an example x, that has been accepted from the rejection sampling process. Similarly, to the marginal $\mathcal{D}_z$, we define $\mathcal{D}_z^f$ to be the marginal distribution of the fine labels z conditional on acceptance. Hence, we can write

$$\mathop{\mathbf{E}}_{(x,z)\sim\mathcal{D}}\Big[f(x)\mathbf{1}\{z = j\}\Big] = \mathop{\mathbf{E}}_{x\sim\mathcal{D}_x}[f(x)]\ \mathop{\mathbf{Pr}}_{z\sim\mathcal{D}_z^f}[z = j]\,. \tag{3}$$

The decomposition of the expectation of Equation (3) is a key step: we now only have to learn the marginal distribution of fine labels conditional on acceptance $\mathcal{D}_z^f$. Since we can draw samples (x, S), it is a straightforward of concentration inequalities, to estimate $\mathbf{E}_{x\sim\mathcal{D}_x}[f(x)]$. Moreover, using Proposition 10 we can learn the distribution $\mathcal{D}_z^f$. For the detailed proof, we refer to the full version of the paper.

3. Any function $f : \mathcal{X} \to [-1,1]$ can be decomposed into $f = f^+ - f^-$ with $f^+, f^- \geq 0$ and, by linearity of expectation, it suffices to work with functions f with image in $[0,1]$.

3.4. Training Models from Coarse Data

Consider a parameterized family of functions $\boldsymbol{x} \to f(\boldsymbol{x}; \boldsymbol{w})$, where the parameters $\boldsymbol{w}$ lie in some parameter space $\mathcal{W} \subseteq \mathbb{R}^p$. For instance, the family may correspond to a feed-forward neural network with L layers. Given a finely labeled training sample $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_N, y_N) \in \mathcal{X} \times \mathcal{Y}$, the parameters $\boldsymbol{w}$ are chosen using a gradient method in order to minimize the empirical risk,

$$\mathcal{L}_N(\boldsymbol{w}) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(\boldsymbol{x}_i; \boldsymbol{w}), y_i) \,,$$

for some loss function $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ and the goal of this optimization task is to minimize the population risk function $\mathcal{L}(\boldsymbol{w}) = \mathbf{E}_{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)}[\ell(f(\boldsymbol{x}; \boldsymbol{w}), y)]$ (where the distribution $\mathcal{D}(\boldsymbol{w}^\star)$ is unknown). For simplicity, let us focus on differentiable loss functions. Performing the SGD algorithm, we can circumvent the lack of knowledge of the population risk function $\mathcal{L}$. Specifically, instead of computing the gradient of $\mathcal{L}(\boldsymbol{w})$, the algorithm steps towards a random direction $\boldsymbol{v}$ with the constraint that the expected value of $\boldsymbol{v}$ is equal to the negative of the true gradient, i.e., it is an unbiased estimate of $-\nabla\mathcal{L}(\boldsymbol{w})$. Such a random vector $\boldsymbol{v}$ can be computed without knowing $\mathcal{D}(\boldsymbol{w}^\star)$ using the interchangeability between the expectation and the gradient operators. Assume that the algorithm is at iteration $t \geq 1$. Let $(\boldsymbol{x}, y) \sim \mathcal{D}(\boldsymbol{w}^\star)$ be a fresh sample and define $\boldsymbol{v}_t$ be the gradient of the loss function with respect to $\boldsymbol{w}$, at the point $\boldsymbol{w}_t$, i.e.,

$$\mathbf{E}[\boldsymbol{v}_t|\boldsymbol{w}_t] = \mathop{\mathbf{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)} [\nabla\ell(f(\boldsymbol{x}; \boldsymbol{w}_t), y)] = \nabla \mathop{\mathbf{E}}_{(\boldsymbol{x},y)\sim\mathcal{D}(\boldsymbol{w}^\star)} [\ell(f(\boldsymbol{x}; \boldsymbol{w}_t), y)] = \nabla\mathcal{L}(\boldsymbol{w}_t) \,.$$

Hence, an algorithm that has query access to a SQ oracle can implement a noisy version of the above iterative process (with inexact gradients, see e.g., d'Aspremont (2008); Devolder et al. (2014); Feldman et al. (2017b)) using the query functions $q_i(\boldsymbol{x}, y) = (\nabla\ell(f(\boldsymbol{x}; \boldsymbol{w}_t), y))_i$ for any $i \in [p]$. Note that the algorithm knows the loss function ℓ, the parameterized functions' family $\{f(\cdot\,; \boldsymbol{w}) : \boldsymbol{w} \in \mathcal{W}\}$ and the current guess $\boldsymbol{w}_t$. Specifically, the algorithm performs p queries (one for each coordinate of the parameter vector) and the oracle returns to the algorithm a noisy gradient vector $\boldsymbol{r}_t$ that satisfies $\|\boldsymbol{r}_t - \nabla\mathcal{L}(\boldsymbol{w}_t)\|_\infty \leq \tau$.

In our setting, we do not have access to the SQ oracle with finely labeled examples. Our main result of this section (Theorem 4) is a mechanism that enables us to obtain access to such an oracle using a few coarsely labeled examples. Hence, we can still perform the noisy gradient descent of the previous paragraph with an additional overhead on the sample complexity, due to the reduction.

4. Learning Gaussians from Coarse Data

In this section, we focus on an unsupervised learning problem with coarse data. Recall that we have already solved such a problem in the discrete setting as an ingredient of our supervised learning result, see Subsection 3.2. In this section, we study the fundamental problem of learning a Gaussian distribution given coarse data. In Subsection 4.1, we show that, under general partitions, this problem is NP-hard. In Subsection 4.2, we show that we can efficiently estimate the Gaussian mean under convex partitions of the space.

4.1. Computational Hardness under General Partitions

In this section, we consider general partitions of the d-dimensional Euclidean space, that may contain non-convex subsets. For instance, a compact convex body and its complement define a

non-convex partition of $\mathbb{R}^d$. In order to get this computational hardness result, we reduce from MAX-CUT and make use of its hardness of approximation (Håstad (2001)). Recall that MAX-CUT can be viewed as a maximization problem, where the objective function corresponds to a particular quadratic function (associated with the Laplacian graph of the given graph instance) and the constraints restrict the solution to lie in the Boolean hypercube (the constraints can be seen geometrically as the intersection of bands, see Figure 2).

We first define MAX-CUT and a variant of MAX-CUT where the optimal cut score is given as part of the input. Let $G = (V, E)$ be a graph[4] with d vertices. A *cut* is a partition of V into two subsets S and $S' = V \setminus S$ and the value of the cut (S, S') is $c(S, S') = \sum_{u,v \in E} \mathbf{1}\{u \in S, v \in S'\}$. The goal of the problem is find the maximum value cut in G, i.e., to partition the vertices into two sets so that the number of edges crossing the cut is maximized. We can define MAX-CUT as the following maximization problem for the graph $G = (V, E)$ with $|V| = d$:

$$\max \sum_{i,j \in E} (\boldsymbol{x}_i - \boldsymbol{x}_j)^2 \,, \quad \text{subj. to} \quad \boldsymbol{x}_i \in \{-1, +1\} \; \forall i \in [d] \,.$$

The objective function is the quadratic $\boldsymbol{x}^T \boldsymbol{L}_G \boldsymbol{x}$, where $\boldsymbol{L}_G$ is the Laplacian matrix of the graph G. We may also assume that the value of the optimal cut is known and is equal to opt.[5] Before proceeding with the overview of the proof, we state a key result of Håstad (2001) about the inapproximability of MAX-CUT .

Lemma 11 (Inapproximability of MaxCut Håstad (2001)) *It is* NP*-hard to approximate* MAX-CUT *to any factor higher than* $16/17$.

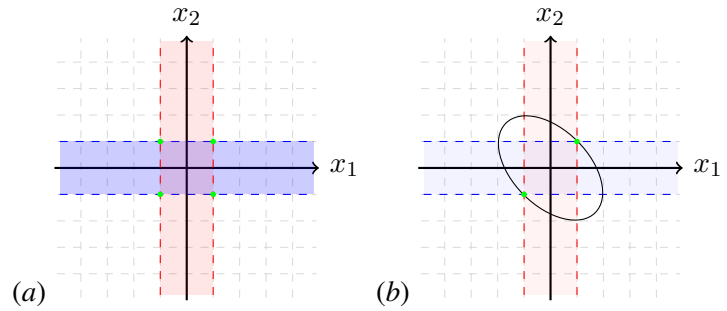

Figure 2: The mixture of partitions corresponding to MAX-CUT .

Sketch of the Proof of Theorem 6 The first step of the proof is to construct the distribution over partitions of $\mathbb{R}^d$. The MAX-CUT problem can be viewed as a collection of $d + 1$ non-convex partitions of the d-dimensional Euclidean space. Consider an instance of MAX-CUT with $|V| = d$ and optimal cut value opt. Consider the collection of $d + 1$ partitions $\mathcal{B} = \{\mathcal{S}_1, \ldots, \mathcal{S}_d, \mathcal{T}\}$. We define the partitions as follows: for any $i = 1, \ldots, d$, we let $S_i = \{\boldsymbol{x} : -1 \leq \boldsymbol{x}_i \leq 1\}$ be the sets that correspond to fat hyperplanes of Figure 2(a) and the partitions $\mathcal{S}_i = \{S_i, S_i^c\}$, i.e., pairs of fat

4. We are going to work with graphs with unit weights.
5. Observe that this problem is still hard, since the maximum value of a cut is bounded by d^2 and, hence, if this problem could be solved efficiently, one would be able to solve MAX-CUT by trying all possible values of opt.

hyperplanes and their complements. These d partitions will simulate the MAX-CUT constraints, i.e., that the solution vector lies in the hypercube $\{-1,1\}^d$. It remains to construct $\mathcal{T}$, which intuitively corresponds to the quadratic objective of MAX-CUT . Fix the covariance matrix $\boldsymbol{\Sigma} = \boldsymbol{L}_G^{-1}\mathrm{opt}$ [6] , i.e., $\boldsymbol{\Sigma}$ is the inverse of the Laplacian normalized by opt. We let $T = \{\boldsymbol{x} : \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q\}$ for some positive value q to be defined later (see Figure 2(b)). Then, we let $\mathcal{T} = \{T, T^c\}$. We construct a mixture π of these partitions by picking each one uniformly at random, i.e., with probability $1/(d+1)$.

Let us assume that there exists an algorithm that, given access to samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$, with *known covariance* $\boldsymbol{\Sigma}$, computes, in time $\mathrm{poly}(d)$, a mean vector $\boldsymbol{\mu}$ so that the output distributions are matched, i.e., $\mathrm{TV}(\mathcal{N}_\pi(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}))$ is upper bounded by $1/d^c$ for some absolute constant $c > 1$. Equivalently this means that the mass that $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ assigns to each set S_i and T is within $\mathrm{poly}(1/d)$ of the corresponding mass that $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$ assigns to the same set. There are two main challenges in order to prove the reduction:

1. How can we generate coarse samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$ since $\boldsymbol{\mu}^\star$ is the solution of the MAX-CUT problem and therefore is unknown?

2. Given opt, is it possible to pick the threshold q of the ellipsoid $T = \{\boldsymbol{x} \in \mathbb{R}^d : \boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q\}$ so that any vector $\boldsymbol{\mu}$ (rounded to belong in $\{-1,1\}^d$), that achieves $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; T) \approx \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T)$ and $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S_i) \approx \mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$, also achieves an approximation ratio better than $16/17$ for the MAX-CUT objective ?

The key observation to answer the first question is that, by the rotation invariance of the Gaussian distribution, the probability $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; T) = \mathbf{Pr}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}\left[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q\right]$ is a constant p that only depends on the value opt of the MAX-CUT problem. Therefore, having this value p, we can flip a coin with this probability and give the coarse sample T if we get heads and T^c otherwise. Similarly, the value of $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$ is an absolute constant that does not depend on $\boldsymbol{\mu}^\star \in \{-1,1\}^d$ and therefore we can again simulate coarse samples by flipping a coin with probability equal to $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma}; S_i)$.

To resolve the second question, we first show that any vector $\boldsymbol{\mu}$ that approximately matches the probabilities of the d fat halfspaces, lies very close to a corner of the hypercube. Therefore, by rounding this guess $\boldsymbol{\mu}$, we obtain exactly a corner of the hypercube without affecting the probability assigned to the ellipsoid constraint by a lot. We then show that any vector of the hypercube that almost matches the probability of the ellipsoid achieves large cut value. In particular, we prove that there exists a value for the threshold q of the ellipsoid $\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q$ that makes the probability $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; T)$ *very sensitive to changes of* $\boldsymbol{\mu}$. Therefore, the only way for the algorithm to match the observed probability is to find a $\boldsymbol{\mu}$ that achieves large cut value. We show the following lemma:

Lemma 12 (Sensitivity of Gaussian Probability of Ellipsoids) *Let $\mathcal{N}(\boldsymbol{\mu}^\star, \boldsymbol{\Sigma})$, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ be d-dimensional Gaussian distributions. Let $\boldsymbol{v}^\star = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}^\star$, $\boldsymbol{v} = \boldsymbol{\Sigma}^{-1/2}\boldsymbol{\mu}$ and assume that $\|\boldsymbol{v}\|_2 \leq \|\boldsymbol{v}^\star\|_2 = 1$. Denote $q = d + \|\boldsymbol{v}^\star\|_2^2 + \sqrt{2d+4}\|\boldsymbol{v}^\star\|_2^2$. Then, assuming d is larger than some sufficiently large absolute constant, it holds that*

$$\left|\mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu}^\star,\boldsymbol{\Sigma})}\left[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q\right] - \mathop{\mathbf{Pr}}_{\boldsymbol{x}\sim\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})}\left[\boldsymbol{x}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{x} \leq q\right]\right| \geq \frac{\|\boldsymbol{v}^\star\|_2^2 - \|\boldsymbol{v}\|_2^2}{6\sqrt{2d+4}} - o(1/\sqrt{d})\,.$$

6. In fact, $\boldsymbol{L}_G$ has zero eigenvalue with eigenvector $(1,\ldots,1)$: we have to project the Laplacian to the subspace orthogonal to $(1,\ldots,1)$ to avoid this. We ignore this technicality here for simplicity.

Notice that with $\boldsymbol{\Sigma} = \boldsymbol{L}_G^{-1}\mathrm{opt}$, in the above lemma, we have $\|\boldsymbol{v}^\star\|_2^2 = 1$, since $\boldsymbol{\mu}^\star$ achieves cut value opt. By assumption, we know that the learning algorithm can find a guess $\boldsymbol{\mu}$ that makes the left hand side of the inequality of Lemma 12 smaller than $\mathrm{poly}(1/d)$. Thus, we obtain that, for d large enough, it must be that $\|\boldsymbol{v}\|_2^2 = \boldsymbol{\mu}^T \boldsymbol{L}_G \boldsymbol{\mu}/\mathrm{opt} \geq 16/17$. Therefore, $\boldsymbol{\mu}$ achieves value greater than $(16/17)\mathrm{opt}$.

Remark 13 *The transformation π used in the above hardness result is not information preserving. In Theorem 6, we prove that it is computationally hard to find a vector $\boldsymbol{\mu} \in \mathbb{R}^d$ that matches in total variation the* observed distribution over coarse labels. *In contrast, as we will see in the upcoming Section 4.2, when the sets of the partitions are convex, we show that there is an efficient algorithm that can solve the same problem and compute some $\boldsymbol{\mu} \in \mathbb{R}^d$ such that $\mathrm{TV}(\mathcal{N}_\pi(\boldsymbol{\mu}^\star), \mathcal{N}_\pi(\boldsymbol{\mu}))$ is small regardless of whether the transformation π is information preserving. When the transformation is information preserving, we can further show that the vector $\boldsymbol{\mu}$ that we compute will be close to $\boldsymbol{\mu}^\star$.*

4.2. Computationally Efficient Mean Estimation under Convex Partitions

In this section, we discuss Theorem 7, which is stated in Section 1. This theorem deals with efficient Gaussian mean estimation in the case of *convex* partitions. In order to prove this result, our strategy is to maximize the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^N \log \mathcal{N}(\boldsymbol{\mu}; S_i)$, where the N (convex) sets $S_1, \ldots, S_N$ are drawn from the coarse Gaussian generative process $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$. The proof of Theorem 7 is decomposed into two structural lemmata, that are stated below. Lemma 14 states that the empirical log-likelihood objective is concave with respect to $\boldsymbol{\mu} \in \mathbb{R}^d$. In order to prove that the Hessian matrix of this objective is negative semi-definite, we use (a variant of) the Brascamp-Lieb inequality for log-concave functions.

Lemma 14 (Concavity of Log-Likelihood (Mean)) *Let $S \subseteq \mathbb{R}^d$ be a convex set. The function $\log \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}; S)$ is concave with respect to the mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$.*

Having established the concavity of the empirical log-likelihood, Lemma 15 comes into play. This lemma states that, given roughly $\widetilde{O}(d/(\epsilon^2\alpha^2))$ samples from $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$, we can guarantee that the maximizer $\widetilde{\boldsymbol{\mu}}$ of the empirical log-likelihood achieves a total variation gap at most ϵ against the true mean vector $\boldsymbol{\mu}^\star$, i.e., $\mathrm{TV}(\mathcal{N}(\widetilde{\boldsymbol{\mu}}), \mathcal{N}(\boldsymbol{\mu}^\star)) \leq \epsilon$. In fact, thanks to the concavity of the empirical log-likelihood objective, it suffices to show that Gaussian distributions $\mathcal{N}(\boldsymbol{\mu})$, that satisfy $\mathrm{TV}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star)) > \epsilon$, will also be significantly sub-optimal solutions of the empirical log-likelihood maximization.

Lemma 15 (Sample Complexity of Empirical Likelihood) *Let $\epsilon, \delta \in (0,1)$ and consider a generative process for coarse d-dimensional Gaussian data $\mathcal{N}_\pi(\boldsymbol{\mu}^\star)$ (see Definition 5). Also, assume that every $\mathcal{S} \in \mathrm{supp}(\pi)$ is a convex partition of the Euclidean space. Let $N = \widetilde{\Omega}(d/(\epsilon^2\alpha^2)\log(1/\delta))$. Consider the empirical log-likelihood objective $\mathcal{L}_N(\boldsymbol{\mu}) = \frac{1}{N}\sum_{i=1}^N \log \mathcal{N}(\boldsymbol{\mu}; S_i)$. Then, with probability at least $1-\delta$, we have that for any Gaussian distribution $\mathcal{N}(\boldsymbol{\mu})$ with $\mathrm{TV}(\mathcal{N}(\boldsymbol{\mu}), \mathcal{N}(\boldsymbol{\mu}^\star)) \geq \epsilon$, it holds that $\max_{\widetilde{\boldsymbol{\mu}} \in \mathbb{R}^d} \mathcal{L}_N(\widetilde{\boldsymbol{\mu}}) - \mathcal{L}_N(\boldsymbol{\mu}) \geq \Omega(\epsilon^2\alpha^2)$.*

Acknowledgments

We thank the anonymous reviewers for useful remarks and comments on the presentation of our manuscript. Dimitris Fotakis and Alkis Kalavasis were supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support Faculty members and Researchers and the procurement of high-cost research equipment grant", project BALSAM, HFRI-FM17-1424. Christos Tzamos and Vasilis Kontonis were supported by the NSF grant CCF-2008006.

References

Javed A Aslam and Scott E Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. *Information and Computation*, 141(2):85–118, 1998.

Ainesh Bakshi, Ilias Diakonikolas, Samuel B Hopkins, Daniel Kane, Sushrut Karmalkar, and Pravesh K Kothari. Outlier-robust clustering of gaussians and other non-spherical mixtures. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 149–159. IEEE Computer Society, 2020.

Maria Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. *Algorithmica*, 72(1):282–315, 2015.

Arnab Bhattacharyya, Rathin Desai, Sai Ganesh Nagarajan, and Ioannis Panageas. Efficient statistics for sparse graphical models from truncated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 1450–1458. PMLR, 2021.

Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1):35–52, 1998.

Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In *Proceedings of the twenty-fourth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 128–138, 2005.

Richard Breen et al. *Regression models: Censored, sample selected, or truncated data*, volume 111. Sage, 1996.

Guy Bukchin, Eli Schwartz, Kate Saenko, Ori Shahar, Rogerio Feris, Raja Giryes, and Leonid Karlinsky. Fine-grained angular contrastive learning with coarse labels. *arXiv preprint arXiv:2012.03515*, 2020.

Moses Charikar, Jacob Steinhardt, and Gregory Valiant. Learning from Untrusted Data. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 47–60, 2017.

Zhuo Chen, Ruizhou Ding, Ting-Wu Chin, and Diana Marculescu. Understanding the impact of label granularity on cnn-based image classification. In *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*, pages 895–904. IEEE, 2018.

Yu Cheng, Ilias Diakonikolas, Rong Ge, and Mahdi Soltanolkotabi. High-dimensional robust mean estimation via gradient descent. In *International Conference on Machine Learning*, pages 1768–1778. PMLR, 2020.

A Clifford Cohen. *Truncated and censored samples: theory and applications*. CRC press, 2016.

Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Efficient Statistics, in High Dimensions, from Truncated Samples. In *59th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 639–649. IEEE, 2018. URL https://arxiv.org/pdf/1809.03986.pdf.

Constantinos Daskalakis, Themis Gouleakis, Christos Tzamos, and Manolis Zampetakis. Computationally and statistically efficient truncated regression. In *Conference on Learning Theory*, pages 955–960. PMLR, 2019.

Constantinos Daskalakis, Dhruv Rohatgi, and Emmanouil Zampetakis. Truncated linear regression in high dimensions. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 10338–10347. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/751f6b6b02bf39c41025f3bcfd9948ad-Paper.pdf.

Alexandre d'Aspremont. Smooth optimization with approximate gradient. *SIAM Journal on Optimization*, 19(3):1171–1183, 2008. URL https://arxiv.org/pdf/math/0512344.pdf.

Jia Deng, Jonathan Krause, and Li Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '13, page 580–587, USA, 2013. IEEE Computer Society. ISBN 9780769549897. doi: 10.1109/CVPR.2013.81. URL https://doi.org/10.1109/CVPR.2013.81.

Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1):37–75, 2014. URL http://www.optimization-online.org/DB_FILE/2010/12/2865.pdf.

I. Diakonikolas, G. Kamath, D. Kane, J. Li, J. Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, pages 1596–1606, 2019.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust Estimators in High Dimensions without the Computational Intractability. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*, pages 655–664, 2016. doi: 10.1109/FOCS.2016.85. URL https://doi.org/10.1109/FOCS.2016.85.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Being Robust (in High Dimensions) Can Be Practical. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, pages 999–1008, 2017a.

Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 73–84. IEEE, 2017b.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robustly Learning a Gaussian: Getting Optimal Error, Efficiently. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 2683–2702, 2018.

Ilias Diakonikolas, Daniel Kane, and Nikos Zarifis. Near-optimal sq lower bounds for agnostically learning halfspaces and relus under gaussian marginals. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 13586–13596. Curran Associates, Inc., 2020a. URL https://proceedings.neurips.cc/paper/2020/file/9d7311ba459f9e45ed746755a32dcd11-Paper.pdf.

Ilias Diakonikolas, Daniel M Kane, Vasilis Kontonis, and Nikos Zarifis. Algorithms and sq lower bounds for pac learning one-hidden-layer relu networks. In *Conference on Learning Theory*, pages 1514–1539. PMLR, 2020b.

John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. *Mathematical Programming*, 114(1):101–114, 2008.

V. Feldman, W. Perkins, and S. Vempala. On the complexity of random satisfiability problems with planted solutions. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC, 2015*, pages 77–86, 2015.

Vitaly Feldman. A general characterization of the statistical query complexity. In *Conference on Learning Theory*, pages 785–830. PMLR, 2017.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh S Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *Journal of the ACM (JACM)*, 64(2): 1–37, 2017a.

Vitaly Feldman, Cristóbal Guzmán, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1265–1277, 2017b. URL https://arxiv.org/pdf/1512.09170.pdf.

Dimitris Fotakis, Alkis Kalavasis, and Christos Tzamos. Efficient parameter estimation of truncated boolean product distributions. In *Conference on Learning Theory*, pages 1586–1600. PMLR, 2020.

Jerome Friedman, Trevor Hastie, Robert Tibshirani, et al. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

Richard D Gill, Mark J Van Der Laan, and James M Robins. Coarsening at random: Characterizations, conjectures, counter-examples. In *Proceedings of the First Seattle Symposium in Biostatistics*, pages 255–294. Springer, 1997.

Surbhi Goel, Aravind Gollakota, Zhihan Jin, Sushrut Karmalkar, and Adam Klivans. Superpolynomial lower bounds for learning one-layer neural networks using gradient descent. In *International Conference on Machine Learning*, pages 3587–3596. PMLR, 2020a.

Surbhi Goel, Aravind Gollakota, and Adam Klivans. Statistical-query lower bounds via functional gradients. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 2147–2158. Curran Associates, Inc., 2020b. URL https://proceedings.neurips.cc/paper/2020/file/17257e81a344982579af1ae6415a7b8c-Paper.pdf.

Christian Gourieroux. *Econometrics of qualitative dependent variables*. Cambridge university press, 2000.

Yanming Guo, Yu Liu, Erwin M Bakker, Yuanhao Guo, and Michael S Lew. Cnn-rnn: a large-scale hierarchical image classification framework. *Multimedia tools and applications*, 77(8):10251–10271, 2018.

Johan Håstad. Some optimal inapproximability results. *Journal of the ACM (JACM)*, 48(4):798–859, 2001. URL http://www.cs.umd.edu/~gasarch/BLOGPAPERS/max3satl.pdf.

Samuel B Hopkins and Jerry Li. How Hard is Robust Mean Estimation? In *Conference on Learning Theory*, pages 1649–1682, 2019.

Peter J Huber. *Robust statistics*, volume 523. John Wiley & Sons, 2004.

Andrew Ilyas, Emmanouil Zampetakis, and Constantinos Daskalakis. A theoretical and practical framework for regression and classification from truncated samples. In *International Conference on Artificial Intelligence and Statistics*, pages 4463–4473. PMLR, 2020.

Qihan Jiao, Zhi Liu, Linwei Ye, and Yang Wang. Weakly labeled fine-grained classification with hierarchy relationship of fine and coarse labels. *Journal of Visual Communication and Image Representation*, 63:102584, 2019.

Qihan Jiao, Zhi Liu, Gongyang Li, Linwei Ye, and Yang Wang. Fine-grained image classification with coarse and fine labels on one-shot learning. In *2020 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, pages 1–6. IEEE, 2020.

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

A. R. Klivans, P. K. Kothari, and R. Meka. Efficient algorithms for outlier-robust regression. In *Conference On Learning Theory, COLT 2018*, pages 1420–1430, 2018.

Vasilis Kontonis, Christos Tzamos, and Manolis Zampetakis. Efficient Truncated Statistics with Unknown Truncation. In *260th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 1578–1595. IEEE, 2019.

Kevin A. Lai, Anup B. Rao, and Santosh Vempala. Agnostic Estimation of Mean and Covariance. In *IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 665–674, 2016.

Jie Lei, Zhenyu Guo, and Yang Wang. Weakly supervised image classification with coarse and fine labels. In *2017 14th Conference on Computer and Robot Vision (CRV)*, pages 240–247. IEEE, 2017.

Gangadharrao S Maddala. *Limited-dependent and qualitative variables in econometrics*. Number 3. Cambridge university press, 1986.

Sai Ganesh Nagarajan and Ioannis Panageas. On the Analysis of EM for truncated mixtures of two Gaussians. In *31st International Conference on Algorithmic Learning Theory (ALT)*, pages 955–960, 2019.

Art Owen et al. Empirical likelihood ratio confidence regions. *The Annals of Statistics*, 18(1): 90–120, 1990.

Art B Owen. Empirical likelihood ratio confidence intervals for a single functional. *Biometrika*, 75 (2):237–249, 1988.

Art B Owen. *Empirical likelihood*. CRC press, 2001.

Zengyi Qin, Jiansheng Chen, Zhenyu Jiang, Xumin Yu, Chunhua Hu, Yu Ma, Suhua Miao, and Rongsong Zhou. Learning fine-grained estimation of physiological states from coarse-grained labels by distribution restoration. *Scientific Reports*, 10(1):1–10, 2020.

M. Ristin, J. Gall, M. Guillaumin, and L. Van Gool. From categories to subcategories: Large-scale image classification with partial class label refinement. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 231–239, 2015. doi: 10.1109/CVPR.2015.7298619.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. URL http://arxiv.org/abs/1409.0575.

Helmut Schneider. *Truncated and censored samples from normal populations*. Marcel Dekker, Inc., 1986.

Ohad Shamir. Distribution-specific hardness of learning neural networks. *The Journal of Machine Learning Research*, 19(1):1135–1163, 2018.

Fariborz Taherkhani, Hadi Kazemi, Ali Dabouei, Jeremy Dawson, and Nasser M Nasrabadi. A weakly supervised fine label classifier enhanced by coarse supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6459–6468, 2019.

David R Thomas and Gary L Grunkemeier. Confidence interval estimation of survival probabilities for censored data. *Journal of the American Statistical Association*, 70(352):865–871, 1975.

James Tobin. Estimation of relationships for limited dependent variables. *Econometrica: journal of the Econometric Society*, pages 24–36, 1958.

Hugo Touvron, Alexandre Sablayrolles, Matthijs Douze, Matthieu Cord, and Hervé Jégou. Grafit: Learning fine-grained image representations with coarse labels. *arXiv preprint arXiv:2011.12982*, 2020.

Santosh Vempala and John Wilmes. Gradient descent for one-hidden-layer neural networks: Polynomial convergence and sq lower bounds. In *Conference on Learning Theory*, pages 3115–3117. PMLR, 2019.

MS Wolynetz. Algorithm as 139: Maximum likelihood estimation in a linear model from confined and censored normal data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(2):195–206, 1979.

Shanshan Wu, Alexandros G. Dimakis, and Sujay Sanghavi. Learning distributions generated by one-layer relu networks. In Hanna M. Wallach, Hugo Larochelle, Alina Beygelzimer, Florence d'Alché-Buc, Emily B. Fox, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pages 8105–8115, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a4d41b834ea903526373a9a1ae2ac66e-Abstract.html.