Learning Sums of Independent Random Variables with Sparse Collective Support

Anindya De
EECS Department
Northwestern University
Evanston, IL, USA
Email: de.anindya@gmail.com

Philip M. Long
Google
Mountain View, CA, USA
Email: plong@google.com

Rocco A. Servedio
CS Department
Columbia University
New York, NY, USA
Email: rocco@cs.columbia.edu

***Abstract*—We study the learnability of sums of independent integer random variables given a bound on the size of the union of their supports. For $\mathcal{A} \subset \mathbb{Z}_+$, a *sum of independent random variables with collective support* $\mathcal{A}$ (called an $\mathcal{A}$-sum in this paper) is a distribution $\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$ where the $\mathbf{X}_i$'s are mutually independent (but not necessarily identically distributed) integer random variables with $\cup_i \mathrm{supp}(\mathbf{X}_i) \subseteq \mathcal{A}$.**

We give two main algorithmic results for learning such distributions:

1) **For the case $|\mathcal{A}| = 3$, we give an algorithm for learning $\mathcal{A}$-sums to accuracy ε that uses $\mathrm{poly}(1/\varepsilon)$ samples and runs in time $\mathrm{poly}(1/\varepsilon)$, independent of N and of the elements of $\mathcal{A}$.**
2) **For an arbitrary constant $k \geq 4$, if $\mathcal{A} = \{a_1, ..., a_k\}$ with $0 \leq a_1 < ... < a_k$, we give an algorithm that uses $\mathrm{poly}(1/\varepsilon) \cdot \log\log a_k$ samples (independent of N) and runs in time $\mathrm{poly}(1/\varepsilon, \log a_k)$.**

We prove an essentially matching lower bound: if $|\mathcal{A}| = 4$, then any algorithm must use $\Omega(\log\log a_4)$ samples even for learning to constant accuracy. We also give similar-in-spirit (but quantitatively very different) algorithmic results, and essentially matching lower bounds, for the case in which $\mathcal{A}$ is not known to the learner.

Our learning algorithms employ new limit theorems which may be of independent interest. Our lower bounds rely on equidistribution type results from number theory. Our algorithms and lower bounds together settle the question of how the sample complexity of learning sums of independent integer random variables scales with the elements in the union of their supports, both in the known-support and unknown-support settings. Finally, all our algorithms easily extend to the "semi-agnostic" learning model, in which training data is generated from a distribution that is only $c\varepsilon$-close to some $\mathcal{A}$-sum for a constant $c > 0$.

***Keywords*-distribution learning; sums of independent random variables; central limit theorems; unsupervised learning; sample complexity**

I. Introduction

The theory of sums of independent random variables forms a rich strand of research in probability. Indeed, many of the best-known and most influential results in probability theory are about such sums; prominent examples include the weak and strong law of large numbers, a host of central limit theorems, and (the starting point of) the theory of large deviations. Within computer science, the well-known "Chernoff-Hoeffding" bounds — i.e., large deviation bounds for sums of independent random variables — are a ubiquitous tool of great utility in many contexts. Not surprisingly, there are several books [1], [2], [3], [4], [5], [6] devoted to the study of sums of independent random variables.

Given the central importance of sums of independent random variables both within probability theory and for a host of applications, it is surprising that even very basic questions about *learning* these distributions were not rigorously investigated until very recently. The problem of learning probability distributions from independent samples has attracted a great deal of attention in theoretical computer science for almost two decades (see [7], [8], [9], [10], [11], [12], [13] and a host of more recent papers), but most of this work has focused on other types of distributions such as mixtures of Gaussians, hidden Markov models, etc. While sums of independent random variables may seem to be a very simple type of distribution, as we shall see below the problem of learning such distributions turns out to be surprisingly tricky.

Before proceeding further, let us recall the standard PAC-style model for learning distributions that was essentially introduced in [7] and that we use in this work. In this model the unknown target distribution $\mathbf{X}$ is assumed to belong to some class $\mathcal{C}$ of distributions. A learning algorithm has access to i.i.d. samples from $\mathbf{X}$, and must produce an efficiently samplable description of a hypothesis distribution $\mathbf{H}$ such that with probability at least (say) $9/10$, the total variation distance $d_{\mathrm{TV}}(\mathbf{X}, \mathbf{H})$ between $\mathbf{X}$ and $\mathbf{H}$ is at most ε. (In the language of statistics, this task is usually referred to as *density estimation*, as opposed to *parametric estimation* in which one seeks to approximately identify the parameters of the unknown distribution $\mathbf{X}$ when $\mathcal{C}$ is a parametric class like Gaussians or mixtures of Gaussians.) In fact, all our positive results hold for the more challenging *semi-agnostic* variant of this model, which is as above except that the assumption that $\mathbf{X} \in \mathcal{C}$ is weakened to the requirement $d_{\mathrm{TV}}(\mathbf{X}, \mathbf{X}^*) \leq c\varepsilon$ for some constant c and some $\mathbf{X}^* \in \mathcal{C}$.

2575-8454/18/$31.00 ©2018 IEEE
DOI 10.1109/FOCS.2018.00036

Learning sums of independent random variables: Formulating the problem. To motivate our choice of learning problem it is useful to recall some relevant context. Recent years have witnessed many research works in theoretical computer science studying the learnability and testability of discrete probability distributions (see e.g. [14], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]); our paper belongs to this line of research. A folklore result in this area is that a simple brute-force algorithm can learn *any* distribution over an M-element set using $\Theta(M/\varepsilon^2)$ samples, and that this is best possible if the distribution may be arbitrary. Thus it is of particular interest to learn classes of distributions over M elements for which a sample complexity dramatically better than this "trivial bound" (ideally scaling as $\log M$, or even independent of M altogether) can be achieved.

This perspective on learning, along with a simple result which we now describe, strongly motivates considering sums of random variables which have small *collective support*. Consider the following very simple learning problem: Let $\{\mathbf{X}_i\}_{i=1}^n$ be independent random variables where $\mathbf{X}_i$ is promised to be supported on the two-element set $\{0, i\}$ but $\mathbf{Pr}[\mathbf{X}_i = i]$ is unknown: what is the sample complexity of learning $\mathbf{X} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$? Even though each random variable $\mathbf{X}_i$ is "as simple as a non-trivial random variable can be" — supported on just two values, one of which is zero — a straightforward lower bound given in [15] shows that any algorithm for learning $\mathbf{X}$ even to constant accuracy must use $\Omega(N)$ samples, which is not much better than the trivial brute-force algorithm based on support size.

Given this lower bound, it is natural to restrict the learning problem by requiring the random variables $\mathbf{X}_1, \ldots, \mathbf{X}_N$ to have small *collective support*, i.e. the union $\mathrm{supp}(\mathbf{X}_1) \cup \cdots \cup \mathrm{supp}(\mathbf{X}_N)$ of their support sets is small. Inspired by this, Daskalakis *et al.* [15] studied the simplest non-trivial version of this learning problem, in which each $\mathbf{X}_i$ is a Bernoulli random variable (so the union of all supports is simply $\{0, 1\}$; note, though, that the $\mathbf{X}_i$'s may have distinct and arbitrary biases). The main result of [15] is that this class (known as *Poisson Binomial Distributions*) can be learned to error ε with $\mathrm{poly}(1/\varepsilon)$ samples — so, perhaps unexpectedly, the complexity of learning this class is completely independent of N, the number of summands. The proof in [15] relies on several sophisticated results from probability theory, including a discrete central limit theorem from [27] (proved using Stein's method) and a "moment matching" result due to Roos [28]. (A subsequent sharpening of the [15] result in [29], giving improved time and sample complexities, also employed sophisticated tools, namely Fourier analysis and algebraic geometry.)

Motivated by this first success, there has been a surge of recent work which studies the learnability of sums of richer classes of random variables. In particular, [16] considered a generalization of [15] in which each $\mathbf{X}_i$ is supported on the set $\{0, 1, \ldots, k-1\}$, and [30] considered a vector-valued generalization in which each $\mathbf{X}_i$ is supported on the set $\{e_1, \ldots, e_k\}$, the standard basis unit vectors in $\mathbb{R}^k$. We will elaborate on these results shortly, but here we first highlight a crucial feature shared by all these results; in all of [15], [16], [30] the collective support of the individual summands forms a "nice and simple" set (either $\{0, 1\}$, $\{0, 1, \ldots, k-1\}$, or $\{e_1, \ldots, e_k\}$). Indeed, the technical workhorses of all these results are various central limit theorems which crucially exploit the simple structure of these collective support sets. (These central limit theorems have since found applications in other settings, such as the design of algorithms for approximating equilibrium [26], [30], [25], [31] as well as stochastic optimization [32].)

In this paper we go beyond the setting in which the collective support of $\mathbf{X}_1, \ldots, \mathbf{X}_N$ is a "nice" set, by studying the learnability of $\mathbf{X}_1 + \cdots + \mathbf{X}_N$ where the collective support may be an *arbitrary* set of non-negative integers. Two questions immediately suggest themselves:

1) How (if at all) does the sample complexity depend on the elements in the common support?
2) Does knowing the common support set help the learning algorithm — how does the complexity vary depending on whether or not the learning algorithm knows the common support?

In this paper we give essentially complete answers to these questions. The answers to these questions emerge from the interface of probability theory and number theory: our algorithms rely on new central limit theorems for sums of independent random variables which we establish, while our matching lower bounds exploit delicate properties of continued fractions and sophisticated equidistribution results from analytic number theory. The authors find it surprising that these two disparate sets of techniques "meet up" to provide matching upper and lower bounds on sample complexity.

We now formalize the problem that we consider.

Our learning problem. Let $\mathbf{X}_1, \ldots, \mathbf{X}_N$ be independent (but not necessarily identically distributed) random variables. Let $\mathcal{A} = \cup_i \mathrm{supp}(\mathbf{X}_i)$ be the union of their supports and assume w.l.o.g. that $\mathcal{A} = \{a_1, ..., a_k\}$ for $a_1 < a_2 < \cdots < a_k \in \mathbb{Z}_{\geq 0}$. Let $\mathbf{S}$ be the sum of these independent random variables, $\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$. We refer to such a random variable $\mathbf{S}$ as an $\mathcal{A}$*-sum*.

We study the problem of learning a unknown $\mathcal{A}$-sum $\mathbf{S}$, given access to i.i.d. draws from $\mathbf{S}$. $\mathcal{A}$-sums generalize several classes of distributions which have recently been intensively studied in unsupervised learning [15], [16], [24], namely Poisson Binomial Distributions and "k-SIIRVs," and are closely related to other such distributions [25], [26] (k-Poisson Multinomial Distributions). These previously studied classes of distributions have all been shown to have learning algorithms with sample complexity $\mathrm{poly}(1/\varepsilon)$ for all constant k.

In contrast, in this paper we show that the picture is more varied for the sample complexity of learning when $\mathcal{A}$ can be any finite set. Roughly speaking (we will give more details soon), two of our main results are as follows:

- *Any* $\mathcal{A}$-sum with $|\mathcal{A}| = 3$ is learnable from $\mathrm{poly}(1/\varepsilon)$ samples independent of N and of the elements of $\mathcal{A}$. This is a significant (and perhaps unexpected) generalization of the efficient learnability of Poisson Binomial Distributions, which corresponds to the case $|\mathcal{A}| = 2$.
- No such guarantee is possible for $|\mathcal{A}| = 4$: if N is large enough, there are infinitely many sets $\mathcal{A} = \{a_1, a_2, a_3, a_4\}$ with $0 \leq a_1 < ... < a_4$ such that $\Omega(\log\log a_4)$ examples are needed even to learn to constant accuracy (for a small absolute constant).

Before presenting our results in more detail, to provide context we recall relevant previous work on learning related distributions.

A. *Previous work*

A *Poisson Binomial Distribution of order* N, or PBD$_N$, is a sum of N independent (not necessarily identical) Bernoulli random variables, i.e. an $\mathcal{A}$-sum for $\mathcal{A} = \{0, 1\}$. Efficient algorithms for learning PBD$_N$ distributions were given in [33], [29], which gave learning algorithms using $\mathrm{poly}(1/\varepsilon)$ samples and $\mathrm{poly}(1/\varepsilon)$ runtime, independent of N.

Generalizing a PBD$_N$ distribution, a k-SIIRV$_N$ *(Sum of Independent Integer Random Variables)* is a $\mathcal{A}$-sum for $\mathcal{A} = \{0, ..., k-1\}$. Daskalakis et al. [16] (see also [24]) gave $\mathrm{poly}(k, 1/\varepsilon)$-time and sample algorithms for learning any k-SIIRV$_N$ distribution to accuracy ε, independent of N.

Finally, a different generalization of PBDs is provided by the class of (N, k)-*Poisson Multinomial Distributions*, or k-PMD$_N$ distributions. Such a distribution is $\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$ where the $\mathbf{X}_i$'s are independent (not necessarily identical) k-dimensional vector-valued random variables each supported on $\{e_1, \ldots, e_k\}$, the standard basis unit vectors in $\mathbb{R}^k$. Daskalakis et al. [30] gave an algorithm that learns any unknown k-PMD$_N$ using $\mathrm{poly}(k/\varepsilon)$ samples and running in time $\min\{2^{O(k^{O(k)} \cdot \log^{O(k)}(1/\varepsilon))}, 2^{\mathrm{poly}(k/\varepsilon)}\}$; this result was subsequently sharpened in [25], [26].

Any $\mathcal{A}$-sum with $|\mathcal{A}| = k$ has an associated underlying k-PMD$_N$ distribution: if $\mathcal{A} = \{a_1, ..., a_k\}$, then writing $\bar{a}$ for the vector $(a_1, \ldots, a_k) \in \mathbb{Z}^k$, an $\mathcal{A}$-sum $\mathbf{S}'$ is equivalent to $\bar{a} \cdot \mathbf{S}$ where $\mathbf{S}$ is an k-PMD$_N$, as making a draw from $\mathbf{S}'$ is equivalent to making a draw from $\mathbf{S}$ and outputting its inner product with the vector $\bar{a}$. However, this does *not* mean that the [30] learning result for k-PMD$_N$ distributions implies a corresponding learning result for $\{a_1, ..., a_k\}$-sums. If an $\mathcal{A}$-sum learning algorithm *were given draws from the underlying* k*-PMD*$_N$, then of course it would be straightforward to run the [30] algorithm, construct a high-accuracy hypothesis distribution $\mathbf{H}$ over $\mathbb{R}^k$, and output $\bar{a} \cdot \mathbf{H}$ as the hypothesis distribution for the unknown $\mathcal{A}$-sum. But when learning $\mathbf{S}'$, the algorithm does not receive draws from the underlying k-PMD$_N$ $\mathbf{S}$; instead it only receives draws from $\bar{a} \cdot \mathbf{S}$. In fact, as we discuss below, this more limited access causes a crucial *qualitative* difference in learnability, namely an inherent dependence on the a_i's in the necessary sample complexity once $k \geq 4$. (The challenge to the learner arising from the blending of the contributions to a $\mathcal{A}$-sum is roughly analogous to the challenge that arises in learning a DNF formula; if each positive example in a DNF learning problem were annotated with an identifier for a term that it satisfies, learning would be trivial.)

B. *The questions we consider and our algorithmic results.*

As detailed above, previous work has extensively studied the learnability of PBDs, k-SIIRVs, and k-PMDs; however, we believe that the current work is the first to study the learnability of general $\mathcal{A}$-sums. A first simple observation is that since any $\mathcal{A}$-sum with $|\mathcal{A}| = 2$ is a scaled and translated PBD, the results on learning PBDs mentioned above easily imply that the sample complexity of learning any $\{a_1, a_2\}$-sum is $\mathrm{poly}(1/\varepsilon)$, independent of the number of summands N and the values a_1, a_2. A second simple observation is that any $\{a_1, ..., a_k\}$-sum with $0 \leq a_1 < ... < a_k$ can be learned using $\mathrm{poly}(a_k, 1/\varepsilon)$ samples, simply by viewing it as an a_k-SIIRV$_N$. But this bound is in general quite unsatisfying – indeed, for large a_k it could be even larger than the trivial $O(N^k/\varepsilon^2)$ upper bound that holds since any $\mathcal{A}$-sum with $|\mathcal{A}| = k$ is supported on a set of size $O(N^k)$.

Once $k \geq 3$ there can be non-trivial additive structure present in the set of values $a_1, \ldots, a_k$. This raises a natural question: is $k = 2$ the only value for which $\mathcal{A}$-sums are learnable from a number of samples that is independent of the domain elements $a_1, \ldots, a_k$? Perhaps surprisingly, our first main result is an efficient algorithm which gives a negative answer. We show that for $k = 3$, the values of the a_i's don't matter; we do this by giving an efficient learning algorithm (even a semi-agnostic one) for learning $\{a_1, a_2, a_3\}$-sums, whose running time and sample complexity are completely independent of a_1, a_2 and a_3:

Theorem 1 (Learning $\mathcal{A}$-sums with $|\mathcal{A}| = 3$, known support)**.** *There is an algorithm and a positive constant* c *with the following properties: The algorithm is given* N*, an accuracy parameter* $\varepsilon > 0$*, distinct values* $a_1 < a_2 < a_3 \in \mathbb{Z}_{\geq 0}$*, and access to i.i.d. draws from an unknown distribution* $\mathbf{S}^*$ *that has total variation distance at most* $c\varepsilon$ *from an* $\{a_1, a_2, a_3\}$*-sum. The algorithm uses* $\mathrm{poly}(1/\varepsilon)$ *draws from* $\mathbf{S}^*$*, runs in* $\mathrm{poly}(1/\varepsilon)$ *time*[1]*, and with probability at least* $9/10$ *outputs a concise representation of a hypothesis distribution* $\mathbf{H}$ *such that* $d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}^*) \leq \varepsilon$.

We also give an algorithm for $k \geq 4$. More precisely, we show:

[1] Here and throughout we assume a unit-cost model for arithmetic operations $+$, $\times$, $\div$.

Theorem 2 (Learning $\mathcal{A}$-sums, known support)**.** *For any $k \geq 4$, there is an algorithm and a constant $c > 0$ with the following properties: it is given N, an accuracy parameter $\varepsilon > 0$, distinct values $a_1 < \cdots < a_k \in \mathbb{Z}_{\geq 0}$, and access to i.i.d. draws from an unknown distribution $\mathbf{S}^*$ that has total variation distance at most $c\varepsilon$ from some $\{a_1, \ldots, a_k\}$-sum. The algorithm runs in time $(1/\varepsilon)^{2^{O(k^2)}} \cdot (\log a_k)^{\mathrm{poly}(k)}$, uses $(1/\varepsilon)^{2^{O(k^2)}} \cdot \log\log a_k$ samples, and with probability at least $9/10$ outputs a concise representation of a hypothesis distribution $\mathbf{H}$ such that $d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}^*) \leq \varepsilon$.*

In contrast with $k = 3$, our algorithm for general $k \geq 4$ has a sample complexity which depends (albeit doubly logarithmically) on a_k. This is a doubly exponential improvement over the naive $\mathrm{poly}(a_k)$ bound which follows from previous a_k-SIIRV learning algorithms [16], [24].

Secondary algorithmic results: Learning with unknown support. We also give algorithms for a more challenging *unknown-support* variant of the learning problem. In this variant the values $a_1, \ldots, a_k$ are not provided to the learning algorithm, but instead only an upper bound $a_{\max} \geq a_k$ is given. Interestingly, it turns out that the unknown-support problem is significantly different from the known-support problem: as explained below, in the unknown-support variant the dependence on $a_{\max}$ kicks in at a smaller value of k than in the known-support variant, and this dependence is exponentially more severe than in the known-support variant.

Using well-known results from hypothesis selection, it is straightforward to show that upper bounds for the known-support case yield upper bounds in the unknown-support case, essentially at the cost of an additional additive $O(k \log a_{\max})/\varepsilon^2$ term in the sample complexity. This immediately yields the following:

Theorem 3 (Learning with unknown support of size k)**.** *For any $k \geq 3$, there is an algorithm and a positive constant c with the following properties: The algorithm is given N, the value k, an accuracy parameter $\varepsilon > 0$, an upper bound $a_{\max} \in \mathbb{Z}_{\geq 0}$, and access to i.i.d. draws from an unknown distribution $\mathbf{S}^*$ that has total variation distance at most $c\varepsilon$ from an $\mathcal{A}$-sum for $\mathcal{A} = \{a_1, \ldots, a_k\} \subset \mathbb{Z}_{\geq 0}$ where $\max_i a_i \leq a_{\max}$. The algorithm uses $O(k \log a_{\max})/\varepsilon^2 + (1/\varepsilon)^{2^{O(k^2)}} \cdot \log\log a_{\max}$ draws from $\mathbf{S}^*$, runs in $\mathrm{poly}((a_{\max})^k) \cdot$ $(1/\varepsilon)^{2^{O(k^2)}} \cdot (\log a_{\max})^{\mathrm{poly}(k)}$ time, and with probability at least $9/10$ outputs a concise representation of a hypothesis distribution $\mathbf{H}$ such that $d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}^*) \leq \varepsilon$.*

Recall that a $\{a_1, a_2\}$-sum is simply a rescaled and translated PBD_N distribution. Using known results for learning PBDs, it is not hard to show that the $k = 2$ case is easy even with unknown support:

Theorem 4 (Learning with unknown support of size 2)**.** *There is an algorithm and a positive constant c with the following properties: The algorithm is given N, an accuracy parameter $\varepsilon > 0$, an upper bound $a_{\max} \in \mathbb{Z}_{+}$, and access to i.i.d. draws from an unknown distribution $\mathbf{S}^*$ that has total variation distance at most $c\varepsilon$ from an $\{a_1, a_2\}$-sum where $0 \leq a_1 < a_2 \leq a_{\max}$. The algorithm uses $\mathrm{poly}(1/\varepsilon)$ draws from $\mathbf{S}^*$, runs in $\mathrm{poly}(1/\varepsilon)$ time, and with probability at least $9/10$ outputs a concise representation of a hypothesis distribution $\mathbf{H}$ such that $d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}^*) \leq \varepsilon$.*

C. Our lower bounds.

We establish sample complexity lower bounds for learning $\mathcal{A}$-sums that essentially match the above algorithmic results.

Known support. Our first lower bound deals with the known support setting. We give an $\Omega(\log\log a_4)$-sample lower bound for the problem of learning an $\{a_1, ..., a_4\}$-sum for $0 \leq a_1 < a_2 < a_3 < a_4$. This matches the dependence on a_k of our $\mathrm{poly}(1/\varepsilon) \cdot \log\log a_k$ upper bound. More precisely, we show:

Theorem 5 (Lower Bound for Learning $\{a_1, ..., a_4\}$-sums, known support)**.** *Let A be any algorithm with the following properties: algorithm A is given N, an accuracy parameter $\varepsilon > 0$, distinct values $0 \leq a_1 < a_2 < a_3 < a_4 \in \mathbb{Z}$, and access to i.i.d. draws from an unknown $\{a_1, ..., a_4\}$-sum $\mathbf{S}^*$; and with probability at least $9/10$ algorithm A outputs a hypothesis distribution $\tilde{\mathbf{S}}$ such that $d_{\mathrm{TV}}(\tilde{\mathbf{S}}, \mathbf{S}^*) \leq \varepsilon$. Then there are infinitely many quadruples (a_1, a_2, a_3, a_4) such that for sufficiently large N, A must use $\Omega(\log\log a_4)$ samples even when run with ε set to a (suitably small) positive absolute constant.*

This lower bound holds even though the target is exactly an $\{a_1, ..., a_4\}$-sum (i.e. it holds even in the easier non-agnostic setting).

Since Theorem 1 gives a $\mathrm{poly}(1/\varepsilon)$ sample and runtime algorithm independent of the size of the a_i's for $k = 3$, the lower bound of Theorem 5 establishes a phase transition between $k = 3$ and $k = 4$ for the sample complexity of learning $\mathcal{A}$-sums: when $k = 3$ the sample complexity is always independent of the actual set $\{a_1, a_2, a_3\}$, but for $k = 4$ it can grow as $\Omega(\log\log a_4)$ (but no faster).

Unknown support. Our second lower bound deals with the unknown support setting. We give an $\Omega(\log a_{\max})$-sample lower bound for the problem of learning an $\{a_1, a_2, a_3\}$-sum with unknown support $0 \leq a_1 < a_2 < a_3 \leq a_{\max}$, matching the dependence on $a_{\max}$ of our algorithm from Theorem 3. More precisely, we prove:

Theorem 6 (Lower Bound for Learning $\{a_1, a_2, a_3\}$-sums, unknown support)**.** *Let A be any algorithm with the following properties: algorithm A is given N, an accuracy parameter $\varepsilon > 0$, a value $0 < a_{\max} \in \mathbb{Z}$, and access to i.i.d. draws from an unknown $\{a_1, a_2, a_3\}$-sum $\mathbf{S}^*$ where $0 \leq a_1 < a_2 < a_3 \leq a_{\max}$; and A outputs a hypothesis distribution $\tilde{\mathbf{S}}$ which with probability at least $9/10$ satisfies*

$d_{\mathrm{TV}}(\tilde{\mathbf{S}}, \mathbf{S}^*) \leq \varepsilon$. *Then for sufficiently large* N, A *must use* $\Omega(\log a_{\max})$ *samples even when run with* ε *set to a (suitably small) positive absolute constant.*

Taken together with our algorithm from Theorem 4 for the case $k = 2$, Theorem 6 establishes another phase transition, but now between $k = 2$ and $k = 3$, for the sample complexity of learning $\mathcal{A}$-sums when $\mathcal{A}$ is unknown. When $|\mathcal{A}| = 2$ the sample complexity is always independent of the actual set, but for $|\mathcal{A}| = 3$ and $0 \leq a_1 < \ldots < a_3$ it can grow as $\Omega(\log a_3)$ (but no faster).

In summary, taken together the algorithms and lower bounds of this paper essentially settle the question of how the sample complexity of learning sums of independent integer random variables with sparse collective support scales with the elements in the collective support, both in the known-support and unknown-support settings.

Discussion. As described above, for an arbitrary set $\{a_1, \ldots, a_k\}$, the sample complexity undergoes a significant phase transition between $k = 3$ and $k = 4$ in the known-support case and between 2 and 3 in the unknown-support case. In each setting the phase transition is a result of "number-theoretic phenomena" (we explain this more later) which can only occur for the larger number and cannot occur for the smaller number of support elements. We find it somewhat surprising that the sample complexities of these learning problems are determined by number-theoretic properties of the support sets.

Organization. In the next section we give some of the key ideas that underlie our algorithms. See Section III for an overview of the ideas behind our lower bounds. Due to space constraints, this extended abstract just gives an overview of the proofs; a full paper with detailed proofs may be found in [34].

II. Techniques for our algorithms

In this section we give an intuitive explanation of some of the ideas that underlie our algorithms and their analysis. While our learning results are for the semi-agnostic model, for simplicity's sake, we focus on the case in which the target distribution $\mathbf{S}$ is actually an $\mathcal{A}$-sum.

A first question, which must be addressed before studying the algorithmic (running time) complexity of learning $\mathcal{A}$-sums, is to understand the sample complexity of learning them. In fact, in a number of recent works on learning various kinds of of "structured" distributions, just understanding the sample complexity of the learning problem is a major goal that requires significant work [33], [35], [16], [36], [30].

In many of the above-mentioned papers, an upper bound on both sample complexity and algorithmic complexity is obtained via a structural characterization of the distributions to be learned; our work follows a similar conceptual paradigm. To give a sense of the kind of structural characterization that can be helpful for learning, we recall the characterization of SIIRV$_N$ distributions that was obtained in [16] (which is the one most closely related to our work). The main result of [16] shows that if $\mathbf{S}$ is any k-SIIRV$_N$ distribution, then at least one of the following holds:

1) $\mathbf{S}$ is ε-close to being supported on $\mathrm{poly}(k/\varepsilon)$ many integers;
2) $\mathbf{S}$ is ε-close to a distribution $c \cdot \mathbf{Z} + \mathbf{Y}$, where $1 \leq c \leq k-1$, $\mathbf{Z}$ is a discretized Gaussian, $\mathbf{Y}$ is a distribution supported on $\{0, \ldots, c-1\}$, and $\mathbf{Y}, \mathbf{Z}$ are mutually independent.

In other words, [16] shows that a k-SIIRV$_N$ distribution is either close to sparse (supported on $\mathrm{poly}(k/\varepsilon)$ integers), or close to a c-scaled discretized Gaussian convolved with a sparse component supported on $\{0, \ldots, c-1\}$. This leads naturally to an efficient learning algorithm that handles Case (1) above "by brute-force" and handles Case (2) by learning $\mathbf{Y}$ and $\mathbf{Z}$ separately (handling $\mathbf{Y}$ "by brute force" and handling $\mathbf{Z}$ by estimating its mean and variance).

In a similar spirit, in this work we seek a more general characterization of $\mathcal{A}$-sums. It turns out, though, that even when $|\mathcal{A}| = 3$, $\mathcal{A}$-sums can behave in significantly more complicated ways than the k-SIIRV$_N$ distributions discussed above. To be more concrete, let $\mathbf{S}$ be a $\{a_1, a_2, a_3\}$-sum with $0 \leq a_1 < a_2 < a_3$. By considering a few simple examples it is easy to see that there are at least four distinct possibilities for "what $\mathbf{S}$ is like" at a coarse level:

- **Example #1:** One possibility is that $\mathbf{S}$ is essentially sparse, with almost all of its probability mass concentrated on a small number of outcomes (we say that such an $\mathbf{S}$ has "small essential support").
- **Example #2:** Another possibility is that $\mathbf{S}$ "looks like" a discretized Gaussian scaled by $|a_i - a_j|$ for some $1 \leq i < j \leq 3$ (this would be the case, for example, if $\mathbf{S} = \sum_{i=1}^{N} \mathbf{X}_i$ where each $\mathbf{X}_i$ is uniform over $\{a_1, a_2\}$).
- **Example #3:** A third possibility is that $\mathbf{S}$ "looks like" a discretized Gaussian with no scaling (the analysis of [16] shows that this is what happens if, for example, N is large and each $\mathbf{X}_i$ is uniform over $\{a_1 = 6, a_2 = 10, a_3 = 15\}$, since $\gcd(6, 10, 15) = 1$).
- **Example #4:** Finally, yet another possibility arises if, say, a_3 is very large (say $a_3 \approx N^2$) while a_2, a_1 are very small (say $O(1)$), and $\mathbf{X}_1, \ldots, \mathbf{X}_{N/2}$ are each uniform over $\{a_1, a_3\}$ while $\mathbf{X}_{N/2+1}, \ldots, \mathbf{X}_N$ are each supported on $\{a_1, a_2\}$ and $\sum_{i=N/2+1}^{N} \mathbf{X}_i$ has very small essential support. In this case, for large N, $\mathbf{S}$ would (at a coarse scale) "look like" a discretized Gaussian scaled by $a_3 - a_1 \approx N^2$, but zooming in, locally each "point" in the support of this discretized Gaussian would actually be a copy of the distribution $\sum_{i=N/2+1}^{N} \mathbf{X}_i$ which has small essential support.

Given these possibilities for how $\mathbf{S}$ might behave, it should not be surprising that our actual analysis for the case $|\mathcal{A}| = 3$ involves four cases (and the above four examples land in the four distinct cases). The overall learning algorithm "guesses" which case the target distribution belongs to and runs a different algorithm for each one; the guessing step is ultimately eliminated using the standard tool of hypothesis testing from statistics. We stress that while the algorithms for the various cases differ in some details, there are many common elements across their analyses, and the well known *kernel method* for density estimation provides the key underlying core learning routine that is used in all the different cases.

In the following intuitive explanation we first consider the case of $\mathcal{A}$-sums for general finite $|\mathcal{A}|$, and later explain how we sharpen the algorithm and analysis in the case $|\mathcal{A}| = 3$ to obtain our stronger results for that case. Our discussion below highlights a new structural result (roughly speaking, a new limit theorem that exploits both "long-range" and "short-range" shift-invariance) that plays a crucial role in our algorithms.

A. *Learning $\mathcal{A}$-sums with $|\mathcal{A}| = k$*

For clarity of exposition in this intuitive overview we make some simplifying assumptions. First, we make the assumption that the $\mathcal{A}$-sum $\mathbf{S}$ that is to be learned has 0 as one value in its k-element support, i.e. we assume that $\mathbf{S} = \mathbf{X}_1 + \ldots + \mathbf{X}_N$ where the support of each $\mathbf{X}_i$ is contained in the set $\{0, a_1, \ldots, a_{k-1}\}$. In fact, we additionally assume that each $\mathbf{X}_i$ is 0-*moded*, meaning that $\mathbf{Pr}[\mathbf{X}_i = 0] \geq \mathbf{Pr}[\mathbf{X}_i = a_j]$ for all $i \in [N]$ and all $j \in [k-1]$. (Getting rid of this assumption in our actual analysis requires us to work with zero-moded variants of the $\mathbf{X}_i$ distributions that we denote $\mathbf{X}'_i$, supported on $O(k^2)$ values that can be positive or negative, but we ignore this for the sake of our intuitive explanation here.) For $j \in [k-1]$ we define

$$\gamma_j := \sum_{i=1}^{N} \mathbf{Pr}[\mathbf{X}_i = a_j],$$

which can be thought of as the "weight" that $\mathbf{X}_1, \ldots, \mathbf{X}_N$ collectively put on the outcome a_j.

A useful tool: hypothesis testing. To explain our approach it is helpful to recall the notion of hypothesis testing in the context of distribution learning [37]. Informally, given T candidate hypothesis distributions, one of which is ε-close to the target distribution $\mathbf{S}$, a hypothesis testing algorithm uses $O(\varepsilon^{-2} \cdot \log T)$ draws from $\mathbf{S}$, runs in $\mathrm{poly}(T, 1/\varepsilon)$ time, and with high probability identifies a candidate distribution which is $O(\varepsilon)$-close to $\mathbf{S}$. We use this tool in a few different ways. Sometimes we will consider algorithms that "guess" certain parameters from a "small" (size-T) space of possibilities; hypothesis testing allows us to assume that such algorithms guess the right parameters, at the cost of increasing the sample complexity and running time by only small factors. In other settings we will show via a case analysis that one of several different learning algorithms will succeed; hypothesis testing yields a combined algorithm that learns no matter which case the target distribution falls into. (We remark that this tool has been used in many recent works on distribution learning, see e.g. [33], [38], [16].)

Our analysis. Let $t_1 = O_{k,\varepsilon}(1) \ll t_2 = O_{k,\varepsilon}(1) \ll \cdots \ll t_{k-1} = O_{k,\varepsilon}(1)$ be fixed values (the exact values are not important here). Let us reorder $a_1, \ldots, a_{k-1}$ so that the weights $\gamma_1 \leq \cdots \leq \gamma_{k-1}$ are sorted in non-decreasing order. An easy special case for us is that each $\gamma_j \leq t_j$. If this is the case, then $\mathbf{S}$ has small "essential support": in a draw from $\mathbf{S} = \mathbf{X}_1 + \cdots + \mathbf{X}_N$, with very high probability for each $j \in [k-1]$ the number of $\mathbf{X}_i$ that take value a_j is at most $\mathrm{poly}(t_{k-1})$, so w.v.h.p. a draw from $\mathbf{S}$ takes one of at most $\mathrm{poly}(t_{k-1})^k$ values. In such a case it is not difficult to learn $\mathbf{S}$ using $\mathrm{poly}((t_{k-1})^k, 1/\varepsilon) = O_{k,\varepsilon}(1)$ samples (see Fact 24). We henceforth may assume that some $\gamma_j > t_j$.

For ease of understanding it is helpful to first suppose that *every* $j \in [k-1]$ has $\gamma_j > t_j$, and to base our understanding of the general case (that some $j \in [k-1]$ has $\gamma_j > t_j$) off of how this case is handled. (It should be noted, though, that our actual analysis of the main learning algorithm does not distinguish this special case.) So let us suppose that for all $j \in [k-1]$ we have $\gamma_j > t_j$. To analyze the target distribution $\mathbf{S}$ in this case, we consider a multinomial distribution $\mathbf{M} = \mathbf{Y}_1 + \cdots + \mathbf{Y}_N$ defined by independent vector-valued random variables $\mathbf{Y}_i$, supported on $0, \boldsymbol{e}_1, \ldots, \boldsymbol{e}_{k-1} \in \mathbb{Z}^{k-1}$, such that for each $i \in [N]$ and $j \in [k-1]$ we have $\mathbf{Pr}[\mathbf{Y}_i = e_j] = \mathbf{Pr}[\mathbf{X}_i = a_j]$. Note that for the multinomial distribution $\mathbf{M}$ defined in this way we have $(a_1, \ldots, a_{k-1}) \cdot \mathbf{M} = \mathbf{S}$.

Using the fact that each γ_j is "large" (at least t_j), recent results from [26] imply that the multinomial distribution $\mathbf{M}$ is close to a $(k-1)$-dimensional discretized Gaussian whose covariance matrix has all eigenvalues large (working with zero-moded distributions is crucial to obtain this intermediate result). In turn, such a discretized multidimensional Gaussian can be shown to be close to a vector-valued random variable in which each marginal (coordinate) is a (± 1)-weighted sum of *independent* large-variance Poisson Binomial Distributions. It follows that $\mathbf{S} = (a_1, \ldots, a_{k-1}) \cdot \mathbf{M}$ is close to a a weighted sum of $k-1$ signed PBDs. [2] A distribution $\tilde{\mathbf{S}}$ is a weighted sum of $k-1$ signed PBDs if $\tilde{\mathbf{S}} = a_1 \cdot \tilde{\mathbf{S}}_1 + \cdots + a_{k-1} \cdot \tilde{\mathbf{S}}_{k-1}$ where $\tilde{\mathbf{S}}_1, \ldots, \tilde{\mathbf{S}}_{k-1}$ are *independent* signed PBDs; in turn, a signed PBD is a sum of independent random variables each of which is either supported on $\{0, 1\}$ or on $\{0, -1\}$. The $\tilde{\mathbf{S}}$ that $\mathbf{S}$ is close to further has the property that each $\tilde{\mathbf{S}}_i$ has "large" variance (large compared with $1/\varepsilon$).

[2] This is a simplification of what the actual analysis establishes, but it gets across the key ideas.

Given the above analysis, to complete the argument in this case that each $\gamma_j > t_j$ we need a way to learn a weighted sum of signed PBDs $\tilde{\mathbf{S}} = a_1 \cdot \tilde{\mathbf{S}}_1 + \cdots + a_{k-1} \cdot \tilde{\mathbf{S}}_{k-1}$ where each $\tilde{\mathbf{S}}_j$ has large variance. This is done with the aid of a new limit theorem, that we establish for distributions of this form. For a detailed treatment of this limit theorem, please see [34]; here, omitting many details, let us explain what this new limit theorem says in our setting and how it is useful for learning. Suppose w.l.o.g. that $\mathbf{Var}[a_{k-1} \cdot \tilde{\mathbf{S}}_{k-1}]$ contributes at least a $\frac{1}{k-1}$ fraction of the total variance of $\tilde{\mathbf{S}}$. Let MIX denote the set of those $j \in \{1, \ldots, k-2\}$ such that $\mathbf{Var}[\tilde{\mathbf{S}}_j]$ is large compared with a_{k-1}, and let $\mathrm{MIX}' = \mathrm{MIX} \cup \{k-1\}$. The new limit theorem implies that the sum $\sum_{j \in \mathrm{MIX}'} a_j \cdot \tilde{\mathbf{S}}_j$ "mixes," meaning that it is very close (in d_{TV}) to a *single* scaled PBD $a_{\mathrm{MIX}'} \cdot \tilde{\mathbf{S}}_{\mathrm{MIX}'}$ where $a_{\mathrm{MIX}'} = \gcd\{a_j : j \in \mathrm{MIX}'\}$. (We remark that the proof of the limit theorem involves a generalization of the notion of shift-invariance from probability theory [39] and a coupling-based method.)

Given this structural result, it is enough to be able to learn a distribution of the form

$$\mathbf{T} := a_1 \cdot \tilde{\mathbf{S}}_1 + \cdots + a_\ell \cdot \tilde{\mathbf{S}}_\ell + a_{\mathrm{MIX}'} \cdot \tilde{\mathbf{S}}_{\mathrm{MIX}'}$$

for which we now know that $a_{\mathrm{MIX}'} \cdot \tilde{\mathbf{S}}_{\mathrm{MIX}'}$ has at least $\frac{1}{\ell+1}$ of the total variance, and each $\tilde{\mathbf{S}}_j$ for $j \in [\ell]$ has $\mathbf{Var}[\tilde{\mathbf{S}}_j]$ which is "not too large" compared with a_{k-1} (but large compared with $1/\varepsilon$). We show how to learn such a distribution using $O_{k,\varepsilon}(1) \cdot \log\log a_{k-1}$ samples (this is where the $\log\log$ dependence in our overall algorithm comes from). This is done, intuitively, by guessing various parameters that essentially define $\mathbf{T}$, specifically the variances $\mathbf{Var}[\tilde{\mathbf{S}}_1], \ldots, \mathbf{Var}[\tilde{\mathbf{S}}_\ell]$. Since each of these variances is roughly at most a_{k-1} (crucially, the limit theorem allowed us to get rid of the $\tilde{\mathbf{S}}_j$'s that had larger variance), via multiplicative gridding there are $O_{\varepsilon,k}(1) \cdot \log a_{k-1}$ possible values for each candidate variance, and via our hypothesis testing procedure this leads to an $O_{\varepsilon,k}(1) \cdot \log\log a_{k-1}$ number of samples that are used to learn.

We now turn to the general case, that some $j \in [k-1]$ has $\gamma_j > t_j$. Suppose w.l.o.g. that $\gamma_1 \leq t_1, \ldots \gamma_{\ell-1} \leq t_{\ell-1}$ and $\gamma_\ell > t_\ell$ (intuitively, think of $\gamma_1, \ldots, \gamma_{\ell-1}$ as "small" and $\gamma_\ell, \ldots, \gamma_{k-1}$ as "large"). Via an analysis akin to the "Light-Heavy Experiment" analysis of [16], we show that in this case the distribution $\mathbf{S}$ is close to a distribution $\tilde{\mathbf{S}}$ with the following structure: $\tilde{\mathbf{S}}$ is a mixture of at most $\mathrm{poly}(t_{\ell-1})^{k-1}$ many distributions each of which is a different shift of a *single* distribution, call it $\mathbf{S}_{\mathrm{heavy}}$, that falls into the special case analyzed above: all of the relevant parameters $\gamma_\ell, \ldots, \gamma_{k-1}$ are large (at least t_ℓ). Intuitively, having at most $\mathrm{poly}(t_{\ell-1})^{k-1}$ many components in the mixture corresponds to having $\gamma_1, \ldots, \gamma_{\ell-1} < t_{\ell-1}$ and $\ell \leq k-1$, and having each component be a shift of the same distribution $\mathbf{S}_{\mathrm{heavy}}$ follows from the fact that there is a "large gap" between $\gamma_{\ell-1}$ and γ_ℓ.

Thus in this general case, the learning task essentially boils down to learning a distribution that is (close to) a mixture of translated copies of a distribution of the form $\mathbf{T}$ given above. Learning such a mixture of translates is a problem that is well suited to the "kernel method" for density estimation. This method has been well studied in classical density estimation, especially for continuous probability densities (see e.g. [37]), but results of the exact type that we need did not seem to previously be present in the literature. (We believe that ours is the first work that applies kernel methods to learn sums of independent random variables.)

We develop tools for multidimensional kernel based learning that suit our context. At its core, the kernel method approach that we develop allows us to do the following: Given a mixture of r translates of $\mathbf{T}$ and constant-factor approximations to $\gamma_\ell, \ldots, \gamma_{k-1}$, the kernel method allows us to learn this mixture to error $O(\varepsilon)$ using only $\mathrm{poly}(1/\varepsilon^\ell, r)$ samples. Further, this algorithm is robust in the sense that the same guarantee holds even if the target distribution is only $O(\varepsilon)$ close to having this structure (this is crucial for us). We then combine this tool with the ideas described above for learning a $\mathbf{T}$-type distribution, and thereby establishing our general learning result for $\mathcal{A}$-sums with $|\mathcal{A}| \geq 4$.

B. The case $|\mathcal{A}| = 3$

In this subsection we build on the discussion in the previous subsection, specializing to $k = |\mathcal{A}| = 3$, and explain the high-level ideas of how we are able to learn with sample complexity $\mathrm{poly}(1/\varepsilon)$ independent of a_1, a_2, a_3.

For technical reasons (related to zero-moded distributions) there are three relevant parameters $t_1 \ll t_2 \ll t_3 = O_\varepsilon(1)$ in the $k = 3$ case. The easy special case that each $\gamma_j \leq t_j$ is handled as discussed earlier (small essential support). As in the previous subsection, let $\ell \in [3]$ be the least value such that $\gamma_\ell > t_\ell$.

In all the cases $\ell = 1, 2, 3$ the analysis proceeds by considering the Light-Heavy-Experiment as discussed in the preceding subsection, i.e. by approximating the target distribution $\mathbf{S}$ by a mixture $\tilde{\mathbf{S}}$ of shifts of the *same* distribution $\mathbf{S}_{\mathrm{heavy}}$. When $\ell = 3$, the "heavy" component $\mathbf{S}_{\mathrm{heavy}}$ is simply a distribution of the form $q_3 \cdot \mathbf{S}_3$ where $\mathbf{S}_3$ is a signed PBD. Crucially, while learning the distribution $\mathbf{T}$ in the previous subsection involved guessing certain variances (which could be as large as a_k, leading to $\log a_k$ many possible outcomes of guesses and $\log\log a_k$ sample complexity), in the current setting the extremely simple structure of $\mathbf{S}_{\mathrm{heavy}} = q_3 \cdot \mathbf{S}_3$ obviates the need to make $\log a_3$ many guesses. Instead, its variance can be approximated in a simple direct way by sampling just two points from $\mathbf{T}$ and taking their difference; this easily gives a constant-factor approximation to the variance of $\mathbf{S}_3$ with non-negligible probability. This success probability can be boosted by repeating this experiment several times (but the number of times does not depend on the a_i values.) We thus can use the kernel-based learning approach in a sample-efficient way, without any dependence on a_1, a_2, a_3 in the sample complexity.

For clarity of exposition, in the remaining intuitive discussion (of the $\ell = 1, 2$ cases) we only consider a special case: we assume that $\mathbf{S} = a_1 \cdot \mathbf{S}_1 + a_2 \cdot \mathbf{S}_2$ where both $\mathbf{S}_1$ and $\mathbf{S}_2$ are large-variance PBDs (so each random variable $\mathbf{X}_i$ is either supported on $\{0, a_1\}$ or on $\{0, a_2\}$, but not on all three values $0, a_1, a_2$). We further assume, clearly without loss of generality, that $\gcd(a_1, a_2) = 1$. (Indeed, our analysis essentially proceeds by reducing the $\ell = 1, 2$ case to this significantly simpler scenario, so this is a fairly accurate rendition of the true case.) Writing $\mathbf{S}_1 = \mathbf{X}_1 + \ldots + \mathbf{X}_{N_1}$ and $\mathbf{S}_2 = \mathbf{Y}_1 + \ldots + \mathbf{Y}_{N_2}$, by zero-modedness we have that $\mathbf{Pr}[\mathbf{X}_i = 0] \geq \frac{1}{2}$ and $\mathbf{Pr}[\mathbf{Y}_i = 0] \geq \frac{1}{2}$ for all i, so $\mathbf{Var}[\mathbf{S}_j] = \Theta(1) \cdot \gamma_j$ for $j = 1, 2$. We assume w.l.o.g. in what follows that $a_1^2 \cdot \gamma_1 \geq a_2^2 \cdot \gamma_2$, so $\mathbf{Var}[\mathbf{S}]$, which we henceforth denote σ^2, is $\Theta(1) \cdot a_1^2 \cdot \gamma_1$.

We now branch into three separate possibilities depending on the relative sizes of γ_2 and a_1^2. Before detailing these possibilities we observe that using the fact that γ_1 and γ_2 are both large, it can be shown that if we sample two points $s^{(1)}$ and $s^{(2)}$ from $\mathbf{S}$, then with constant probability the value $\frac{|s^{(1)} - s^{(2)}|}{a_1}$ provides a constant-factor approximation to γ_1.

First possibility: $\gamma_2 < \varepsilon^2 \cdot a_1^2$. The algorithm samples two more points $s^{(3)}$ and $s^{(4)}$ from the distribution $\mathbf{S}$. The crucial idea is that with constant probability these two points can be used to obtain a constant-factor approximation to γ_2; we now explain how this is done. For $j \in \{3, 4\}$, let $s^{(j)} = a_1 \cdot s_1^{(j)} + a_2 \cdot s_2^{(j)}$ where $s_1^{(j)} \sim \mathbf{S}_1$ and $s_2^{(j)} \sim \mathbf{S}_2$, and consider the quantity $s^{(3)} - s^{(4)}$. Since γ_2 is so small relative to a_1, the "sampling noise" from $a_1 \cdot s_1^{(3)} - a_1 \cdot s_1^{(4)}$ is likely to overwhelm the difference $a_2 \cdot s_2^{(3)} - a_2 \cdot s_2^{(4)}$ at a "macroscopic" level. The key idea to deal with this is to *analyze the outcomes modulo* a_1. In the modular setting, because $\mathbf{Var}[\mathbf{S}_2] = \Theta(1) \cdot \gamma_2 \ll a_1^2$, one can show that with constant probability $|(a_2^{-1} \cdot (s_2^{(3)} - s_2^{(4)})) \mod a_1|$ is a constant-factor approximation to γ_2. (Note that as a_1 and a_2 are coprime, the operation a_2^{-1} is well defined modulo a_1.) A constant-factor approximation to γ_2 can be used together with the constant-factor approximation to γ_1 to employ the aforementioned "kernel method" based algorithm to learn the target distribution $\mathbf{S}$. The fact that here we can use only two samples (as opposed to $\log \log a_1$ samples) to estimate γ_2 is really the crux of why for the $k = 3$ case, the sample complexity is independent of a_1. (Indeed, we remark that our analysis of the lower bound given by Theorem 5 takes place in the modular setting and this "mod a_1" perspective is crucial for constructing the lower bound examples in that proof.)

Second possibility: $a_1^2/\varepsilon^2 > \gamma_2 > \varepsilon^2 \cdot a_1^2$. Here, by multiplicative gridding we can create a list of $O(\log(1/\varepsilon))$ guesses such that at least one of them is a constant-factor approximation to γ_2. Again, we use the kernel method and the approximations to γ_1 and γ_2 to learn $\mathbf{S}$.

Third possibility: The last possibility is that $\gamma_2 \geq a_1^2/\varepsilon^2$. In this case, we show that $\mathbf{S}$ is in fact ε-close to the discretized Gaussian (with no scaling; recall that $\gcd(a_1, a_2) = 1$) that has the appropriate mean and variance. Given this structural fact, it is easy to learn $\mathbf{S}$ by just estimating the mean and the variance and outputting the corresponding discretized Gaussian. This structural fact follows from our new limit theorem, mentioned earlier; we conclude this section with a discussion of this new limit theorem.

C. Limit theorems.

Here is a simplified version of our new limit theorem, specialized to the case $D = 2$:

Simplified version of limit theorem. *Let* $\mathbf{S} = r_1 \cdot \mathbf{S}_1 + r_2 \cdot \mathbf{S}_2$ *where* $\mathbf{S}_1, \mathbf{S}_2$ *are independent signed PBDs and* r_1, r_2 *are nonzero integers such that* $\gcd(r_1, r_2) = 1$, $\mathbf{Var}[r_1 \cdot \mathbf{S}_1] \geq \mathbf{Var}[r_2 \cdot \mathbf{S}_2]$, *and* $\mathbf{Var}[\mathbf{S}_2] \geq \max\{\frac{1}{\varepsilon^8}, \frac{r_1}{\varepsilon}\}$. *Then* $\mathbf{S}$ *is* $O(\varepsilon)$-*close in total variation distance to a signed PBD* $\mathbf{S}'$ *(and hence to a signed discretized Gaussian) with* $\mathbf{Var}[\mathbf{S}'] = \mathbf{Var}[\mathbf{S}]$.

If a distribution $\mathbf{S}$ is close to a discretized Gaussian in Kolmogorov distance and is $1/\sigma$-shift invariant (i.e. $d_{\mathrm{TV}}(\mathbf{S}, \mathbf{S} + 1) \leq 1/\sigma$), then $\mathbf{S}$ is close to a discretized Gaussian in total variation distance [40], [41]. Gopalan, et al. [42] used a coupling based argument to establish a similar central limit theorem to obtain PRGs for certain space bounded branching programs. Unfortunately, in the setting of the lemma stated above, it is not immediately clear why $\mathbf{S}$ should have $1/\sigma$-shift invariance.

To deal with this, we give a novel analysis exploiting *shift-invariance at multiple different scales*. Roughly speaking, because of the $r_1 \cdot \mathbf{S}_1$ component of $\mathbf{S}$, it can be shown that $d_{\mathrm{TV}}(\mathbf{S}, \mathbf{S} + r_1) = 1/\sqrt{\mathbf{Var}[\mathbf{S}_1]}$, i.e. $\mathbf{S}$ has good "shift-invariance at the scale of r_1"; by the triangle inequality $\mathbf{S}$ is also not affected much if we shift by a small integer multiple of r_1. The same is true for a few shifts by r_2, and hence also for a few shifts by *both* r_1 and r_2. If $\mathbf{S}$ is approximated well by a discretized Gaussian, though, then it is also not affected by small shifts, including shifts by 1, and in fact we need such a guarantee to prove approximation by a discretized Gaussian through coupling. However, since $\gcd(r_1, r_2) = 1$, basic number theory implies that we can achieve any small integer shift via a small number of shifts by r_1 and r_2, and therefore $\mathbf{S}$ has the required "fine-grained" shift-invariance (at scale 1) as well. Intuitively, for this to work we need samples from $r_2 \cdot \mathbf{S}_2$ to "fill in the gaps" between successive values of $r_1 \cdot \mathbf{S}_1$ – this is why we need $\mathbf{Var}[\mathbf{S}_2] \gg r_1$.

This idea of exploiting both long-range and short-range shift invariance is new to the best of our knowledge [41] and seems likely to be of use in proving new central limit theorems.

III. Lower bound techniques

In this section we give an overview of the ideas behind our lower bounds. Both of our lower bounds actually work by considering restricted $\mathcal{A}$-sums: our lower bounds can be proved using only distributions $\mathbf{S}$ of the form $\mathbf{S} = \sum_{i=1}^{k} a_i \cdot \mathbf{S}_i$, where $\mathbf{S}_1, \ldots, \mathbf{S}_k$ are independent PBDs; equivalently, $\mathbf{S} = \sum_{i=1}^{N} \mathbf{X}_i$ where each $\mathbf{X}_i$ is supported on one of $\{0, a_1\}$, $\ldots$, $\{0, a_k\}$.

A useful reduction. The problem of learning a distribution modulo an integer plays a key role in both of our lower bound arguments. More precisely, both lower bounds use a reduction which we establish that an efficient algorithm for learning weighted PBDs with weights $0 < a_1 < \ldots < a_k$ implies an efficient algorithm for learning with weights $a_1, \ldots, a_{k-1}$ modulo a_k. This problem is specified as follows. An algorithm which is given access to i.i.d. draws from the distribution $(\mathbf{S} \mod a_k)$ (note that this distribution is supported over $\{0, 1, \ldots, a_k - 1\}$) where $\mathbf{S}$ is of the form $a_1 \cdot \mathbf{S}_1 + \ldots + a_{k-1} \cdot \mathbf{S}_{k-1}$ and $\mathbf{S}_1, \ldots, \mathbf{S}_{k-1}$ are PBDs. The algorithm should produce a high-accuracy hypothesis distribution for $(\mathbf{S} \mod a_k)$. We stress that the example points provided to the learning algorithm all lie in $\{0, \ldots, a_k - 1\}$ (so certainly any reasonable hypothesis distribution should also be supported on $\{0, \ldots, a_k - 1\}$). Such a reduction is useful for our lower bounds because it enables us to prove a lower bound for learning $\sum_{i=1}^{k} a_i \cdot \mathbf{S}_i$ by proving a lower bound for learning $\sum_{i=1}^{k-1} a_i \cdot \mathbf{S}_i \mod a_k$.

The high level idea of this reduction is fairly simple so we sketch it here. Let $\mathbf{S} = a_1 \cdot \mathbf{S}_1 + \cdots + a_{k-1} \cdot \mathbf{S}_{k-1}$ be a weighted sum of PBDs such that $(\mathbf{S} \mod a_k)$ is the target distribution to be learned and let N be the total number of summands in all of the PBDs. Let $\mathbf{S}_k$ be an independent PBD with mean and variance $\Omega(N^\star)$. The key insight is that by taking $N^\star$ sufficiently large relative to N, the distribution of $(\mathbf{S} \mod a_k) + a_k \cdot \mathbf{S}_k$ (which can easily be simulated by the learner given access to draws from $(\mathbf{S} \mod a_k)$ since it can generate samples from $a_k \cdot \mathbf{S}_k$ by itself) can be shown to be statistically very close to that of $\mathbf{S}' := \mathbf{S} + a_k \cdot \mathbf{S}_k$. Here is an intuitive justification: We can think of the different possible outcomes of $a_k \cdot \mathbf{S}_k$ as dividing the support of $\mathbf{S}'$ into bins of width a_k. Sampling from $\mathbf{S}'$ can be performed by picking a bin boundary (a draw from $a_k \cdot \mathbf{S}_k$) and an offset $\mathbf{S}$. While adding $\mathbf{S}$ may take the sample across multiple bin boundaries, if $\mathbf{Var}[\mathbf{S}_k]$ is sufficiently large, then adding $\mathbf{S}$ typically takes $a_k \cdot \mathbf{S}_k + \mathbf{S}$ across a small fraction of the bin boundaries. Thus, the conditional distribution given membership in a bin is similar between bins that have high probability under $\mathbf{S}'$, which means that all of these conditional distributions are similar to the distribution of $\mathbf{S}' \mod a_k$ (which is a mixture of them). Finally, $\mathbf{S}' \mod a_k$ has the same distribution as $\mathbf{S} \mod a_k$. Thus, given samples from $(\mathbf{S} \mod a_k)$, the learner can essentially simulate samples from $\mathbf{S}'$. However, $\mathbf{S}'$ is is a weighted sum of k PBDs, which by the assumption of our reduction theorem can be learned efficiently. Now, assuming the learner has a hypothesis $\mathbf{H}$ such that $d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}') \leq \varepsilon$, it immediately follows that $d_{\mathrm{TV}}((\mathbf{H} \mod a_k), (\mathbf{S}' \mod a_k)) \leq d_{\mathrm{TV}}(\mathbf{H}, \mathbf{S}') \leq \varepsilon$ as desired.

Proof overview of Theorem 5. At this point we have the task of proving a lower bound for learning weighted PBDs over $\{0, a_1, a_2\}$ mod a_3. We establish such a lower bound using Fano's inequality. To get a sample complexity lower bound of $\Omega(\log \log a_3)$ from Fano's inequality, we must construct $T = \log^{\Omega(1)} a_3$ distributions $\mathbf{S}_1, \ldots, \mathbf{S}_T$, where each $\mathbf{S}_i$ is a weighted PBD on $\{0, a_1, a_2\}$ modulo a_3, meeting the following requirements: $d_{\mathrm{TV}}(\mathbf{S}_i, \mathbf{S}_j) = \Omega(1)$ if $i \neq j$, and $D_{KL}(\mathbf{S}_i || \mathbf{S}_j) = O(1)$ for all $i, j \in T$. In other words, applying Fano's inequality requires us to exhibit a large number of distributions (belonging to the family for which we are proving the lower bound) such that any two distinct distributions in the family are *far* in total variation distance but *close* in terms of KL-divergence. The intuitive reason for these two competing requirements is that if $\mathbf{S}_i$ and $\mathbf{S}_j$ are 2ε-far in total variation distance, then a successful algorithm for learning to error at most ε must be able to distinguish $\mathbf{S}_i$ and $\mathbf{S}_j$. On the other hand, if $\mathbf{S}_i$ and $\mathbf{S}_j$ are close in KL divergence, then it is difficult for any learning algorithm to distinguish between $\mathbf{S}_i$ and $\mathbf{S}_j$.

Now we present the high-level idea of how we may construct distributions $\mathbf{S}_1, \mathbf{S}_2, \ldots$ with the properties described above to establish Theorem 5. The intuitive description of $\mathbf{S}_i$ that we give below does not align perfectly with our actual construction, but this simplified description is hopefully helpful in getting across the main idea.

For the construction we fix $a_1 = 1$, $a_2 = p$ and $a_3 = q$. (We discuss how p and q are selected later; this is a crucial aspect of our construction.) The i-th distribution $\mathbf{S}_i$ is $\mathbf{S}_i = \mathbf{U}_i + p\mathbf{V}_i \mod q$; we describe the distribution $\mathbf{S}_i = \mathbf{U}_i + p\mathbf{V}_i \mod q$ in two stages, first by describing each $\mathbf{V}_i$, and then by describing the corresponding $\mathbf{U}_i$. In the actual construction $\mathbf{U}_i$ and $\mathbf{V}_i$ will be shifted binomial distributions. Since a binomial distribution is rather flat within one standard deviation of its mean, and decays exponentially after that, it is qualitatively somewhat like the uniform distribution over an interval; for this intuitive sketch it is helpful to think of $\mathbf{U}_i$ and $\mathbf{V}_i$ as actually being uniform distributions over intervals. We take the support of $\mathbf{V}_1$ to be an interval of length q/p, so that adjacent members of the support of $(p\mathbf{V}_1 \mod q)$ will be at distance p apart from each other. More generally, taking $\mathbf{V}_i$ to be uniform over an interval of length $2^{i-1}q/p$, the average gap between adjacent members of $\mathrm{supp}(p\mathbf{V}_i \mod q)$ is of length essentially $p/2^{i-1}$, and by a careful choice of p relative to q one might furthermore hope that the gaps would be "balanced", so that they are all of length roughly $p/2^{i-1}$. (This "careful choice" is the technical heart of our actual construction presented later.)

How does $\mathbf{U}_i$ enter the picture? The idea is to take each $\mathbf{U}_i$ to be uniform over a *short* interval, of length $3p/2^i$. This "fills in each gap" and additionally "fills in the first half of the following gap;" as a result, the first half of each gap ends up with twice the probability mass of the second half. (As a result, every two points have probability mass within a constant factor of each other under every distribution — in fact, any point under any one of our distributions has probability mass within a constant factor of that of any other point under any other one of our distributions. This gives the $D_{KL}(\mathbf{S}_i||\mathbf{S}_j) \leq O(1)$ upper bound mentioned above.) For example, recalling that the "gaps" in $\mathrm{supp}(p\mathbf{V}_1 \mod q)$ are of length p, choosing $\mathbf{U}_1$ to be uniform over $\{1, \ldots, 3p/2\}$ will fill in each gap along with the first half of the following gap. Intuitively, each $\mathbf{S}_i = \mathbf{U}_i + p\mathbf{V}_i$ is a "striped" distribution, with equal-width "light stripes" (of uniformly distributed smaller mass) and "dark stripes" (of uniformly distributed larger mass), and each $\mathbf{S}_{i+1}$ has stripes of width half of the $\mathbf{S}_i$-sum's stripes. Roughly speaking, two such distributions $\mathbf{S}_i$ and $\mathbf{S}_j$ "overlap enough" (by a constant fraction) so that they are difficult to distinguish; however they are also "distinct enough" that a successful learning algorithm must be able to distinguish which $\mathbf{S}_i$ its samples are drawn from in order to generate a high-accuracy hypothesis.

We now elaborate on the careful choice of p and q that was mentioned above. The critical part of this choice of p and q is that for $i \geq 1$, in order to get "evenly spaced gaps," the remainders of $p \cdot s$ modulo q where $s \in \{1, \ldots, 2^{i-1}q/p\}$ should be roughly evenly spaced, or *equidistributed*, in the group $\mathbb{Z}_q$. Here the notion of "evenly spaced" is with respect to the "wrap-around" distance (also known as the *Lee metric*) on the group $\mathbb{Z}_q$ (so, for example, the wrap-around distance between 1 and 2 is 1, whereas the wrap-around distance between $q-1$ and 1 is 2). Roughly speaking, we would like $p \cdot s$ modulo q to be equidistributed in $\mathbb{Z}_q$ when $s \in \{1, \ldots, 2^{i-1}q/p\}$, for a range of successive values of i (the more the better, since this means more distributions in our hard family and a stronger lower bound). Thus, qualitatively, we would like the remainders of p modulo q to be *equidistributed at several scales.* We note that equidistribution phenomena are well studied in number theory and ergodic theory, see e.g. [43].

While this connection to equidistribution phenomena is useful for providing visual intuition (at least to the authors), in our attempts to implement the construction using powers of two that was just sketched, it seemed that in order to control the errors that arise in fact a *doubly* exponential growth was required, leading to the construction of only $\Theta(\log\log q)$ such distributions and hence a $\Omega(\log\log\log q)$ sample complexity lower bound. Thus to achieve an $\Omega(\log\log q)$ sample complexity lower bound, our actual choice of p and q comes from the theory of continued fractions. In particular, we choose p and q so that p/q has a continued fraction representation with "many" ($O(\log q)$, though for technical reasons we use only $\log^{\Theta(1)} q$ many) convergents that grow relatively slowly. These $T = \log^{\Theta(1)} q$ convergents translate into T distributions $\mathbf{S}_1, \ldots, \mathbf{S}_T$ in our "hard family" of distributions, and thus into an $\Omega(\log\log q)$ sample lower bound via Fano's inequality.

The key property that we use is a well-known fact in the theory of continued fractions: if g_i/h_i is the i^{th} convergent of a continued fraction for p/q, then $|g_i/h_i - p/q| \leq 1/(h_i \cdot h_{i+1})$. In other words, the i^{th} convergent g_i/h_i provides a non-trivially good approximation of p/q (note that getting an error of $1/h_i$ would have been trivial). From this property, it is not difficult to see that the remainders of $p \cdot \{1, \ldots, h_i\}$ are roughly equidistributed modulo q.

Thus, a more accurate description of our (still idealized) construction is that we choose $\mathbf{V}_i$ to be uniform on $\{1, \ldots, h_i\}$ and $\mathbf{U}_i$ to be uniform on roughly $\{1, \ldots, (3/2) \cdot (q/h_i)\}$. So as to have as many distributions as possible in our family, we would like $h_i \approx (q/p) \cdot c^i$ for some fixed $c > 1$. This can be ensured by choosing p, q such that all the numbers appearing in the continued fraction representation of p/q are bounded by an absolute constant; in fact, in the actual construction, we simply take p/q to be a convergent of $1/\phi$ where ϕ is the golden ratio. With this choice we have that the i^{th} convergent of the continued fraction representation of $1/\phi$ is g_i/h_i, where $h_i \approx ((\sqrt{5}+1)/2)^i$. This concludes our informal description of the choice of p and q.

Again, in our actual construction, we cannot use uniform distributions over intervals (since we need to use PBDs), but rather we have shifted binomial distributions. This adds some technical complication to the formal proofs, but the core ideas behind the construction are indeed as described above.

Proof overview of Theorem 6. As mentioned earlier, Theorem 6 also uses our reduction from the modular learning problem. Taking $a_1 = 0$ and $a_3 \approx a_{\max}$ to be "known" to the learner, we show that any algorithm for learning a distribution of the form $(a_2\mathbf{S}_2 \mod a_3)$, where $0 < a_2 < a_3$ is unknown to the learner and $\mathbf{S}_2$ is a PBD_N, must use $\Omega(\log a_3)$ samples. Like Theorem 5, we prove this using Fano's inequality, by constructing a "hard family" of $(a_3)^{\Omega(1)}$ many distributions of this type such that any two distinct distributions in the family have variation distance $\Omega(1)$ but KL-divergence $O(1)$.

We sketch the main ideas of our construction, starting with the upper bound on KL-divergence. The value a_3 is taken to be a prime. The same PBD_N distribution $\mathbf{S}_2$, which is simply a shifted binomial distribution and may be assumed to be "known" to the learner, is used for all of the distributions in the "hard family", so different distributions in this family differ only in the value of a_2. The shifted binomial distribution $\mathbf{S}_2$ is taken to have variance $\Theta((a_3)^2)$, so, very roughly, $\mathbf{S}_2$ assigns significant probability on $\Theta(a_3)$ distinct values. From this property, it is not difficult to show (similar to our earlier discussion) that any point in the domain $\{0, 1, \ldots, a_3 - 1\}$ under any one of our distributions has probability mass within a constant factor of that of any other point under any other one of our distributions (where the constant factor depends on the hidden constant in the $\Theta((a_3)^2)$). This gives the required $O(1)$ upper bound on KL-divergence.

It remains to sketch the $\Omega(1)$ lower bound on variation distance. As in our discussion of the Theorem 5 lower bound, for intuition it is convenient to think of the shifted binomial distribution $\mathbf{S}_2$ as being uniform over an interval of the domain $\{0, 1, \ldots, a_3 - 1\}$; by carefully choosing the variance and offset of this shifted binomial, we may think of this interval as being $\{0, 1, \ldots, r - 1\}$ for $r = \kappa a_3$ for some small constant $\kappa > 0$ (the constant κ again depends on the hidden constant in the $\Theta((a_3)^2)$) value of the variance). So for the rest of our intuitive discussion we view the distributions in the hard family as being of the form $(a_2 \cdot \mathbf{U}_r \mod a_3)$ where $\mathbf{U}_r$ is uniform over $\{0, 1, \ldots, r - 1\}$, $r = \kappa a_3$.

Recalling that a_3 is prime, it is clear that for any $0 < a_2 < a_3$, the distribution $(a_2 \cdot \mathbf{U}_r \mod a_3)$ is uniform over an $(r = \kappa a_3)$-element subset of $\{0, \ldots, a_3 - 1\}$. If a_2 and a_2' are two independent uniform random elements from $\{1, \ldots, a_3 - 1\}$, then since κ is a small constant, intuitively the overlap between the supports of $(a_2 \cdot \mathbf{U}_r \mod a_3)$ and $(a_2' \cdot \mathbf{U}_r \mod a_3)$ should be small, and consequently the variation distance between these two distributions should be large. This in turn suggests that by drawing a large random set of values for a_2, it should be possible to obtain a large family of distributions of the form $(a_2 \cdot \mathbf{U}_r \mod a_3)$ such that any two of them have large variation distance. We make this intuition precise using a number-theoretic equidistribution result of Shparlinski [44] and a probabilistic argument showing that indeed a random set of $(a_3)^{1/3}$ choices of a_2 is likely to have the desired property. This gives a "hard family" of size $(a_3)^{1/3}$, leading to an $\Omega(\log a_3) = \Omega(\log a_{\max})$ lower bound via Fano's inequality. As before some technical work is required to translate these arguments for the uniform distribution over to the shifted binomial distributions that we actually have to work with.

Acknowledgement

R.A.S. was supported by NSF awards CCF-1420349 and CCF-1563155. A.D. was supported by a start-up grant from Northwestern University and NSF award CCF-1814706.

References

[1] B. Gnedenko and A. Kolmogorov, *Independent Random Variables*. Cambridge, Massachusetts: Addison-Wesley, 1954. 1

[2] V. V. Petrov, *Sums of Independent Random Variables*. Springer-Verlag Berlin Heidelberg, 1975, translated by A.A. Brown. 1

[3] V. Petrov, *Limit Theorems of Probability Theory: Sequences of Independent Random Variables*. Oxford University Press, 1995. 1

[4] Y. V. Prokhorov and V. Statulevicius, *Limit Theorems of Probability Theory*, 2000, vol. 71. 1

[5] O. Klesov, *Limit Theorems for Multi-Indexed Sums of Random Variables*, 01 2014, vol. 71. 1

[6] A. Borovkov and A. Balakrishnan, *Advances in probability theory: limit theorems for sums of random variables*, ser. Trudy Instituta matematiki, 1985. 1

[7] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. Schapire, and L. Sellie, "On the learnability of discrete distributions," in *Proceedings of the 26th Symposium on Theory of Computing*, 1994, pp. 273–282. 1

[8] S. Dasgupta, "Learning mixtures of Gaussians," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, 1999, pp. 634–644. 1

[9] S. Arora and R. Kannan, "Learning mixtures of arbitrary Gaussians," in *Proceedings of the 33rd Symposium on Theory of Computing*, 2001, pp. 247–257. 1

[10] S. Vempala and G. Wang, "A spectral algorithm for learning mixtures of distributions," in *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, 2002, pp. 113–122. 1

[11] A. T. Kalai, A. Moitra, and G. Valiant, "Efficiently learning mixtures of two Gaussians," in *STOC*, 2010, pp. 553–562. 1

[12] A. Moitra and G. Valiant, "Settling the polynomial learnability of mixtures of Gaussians," in *FOCS*, 2010, pp. 93–102. 1

[13] M. Belkin and K. Sinha, "Polynomial learning of distribution families," in *FOCS*, 2010, pp. 103–112. 1

[14] C. Daskalakis, I. Diakonikolas, and R. Servedio, "Learning k-modal distributions via testing," in *SODA*, 2012, pp. 1371–1385. 2

[15] ——, "Learning Poisson Binomial Distributions," in *Proceedings of the 44th Symposium on Theory of Computing*, 2012, pp. 709–728. 2

[16] C. Daskalakis, I. Diakonikolas, R. O'Donnell, R. Servedio, and L.-Y. Tan, "Learning Sums of Independent Integer Random Variables," in *FOCS*, 2013, pp. 217–226. 2, 3, 4, 5, 6, 7

[17] Y. Rabani, L. J. Schulman, and C. Swamy, “Learning mixtures of arbitrary distributions over large discrete domains,” in *Innovations in Theoretical Computer Science, ITCS 2014*, 2014, pp. 207–224. 2

[18] J. Acharya, C. Daskalakis, and G. Kamath, “Optimal testing for properties of distributions,” in *Advances in Neural Information Processing Systems 28 (NIPS)*, 2015, pp. 3591–3599. 2

[19] J. Acharya and C. Daskalakis, “Testing poisson binomial distributions,” in *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms,SODA 2015*, 2015, pp. 1829–1840. 2

[20] C. L. Canonne, “Big data on the rise? - testing monotonicity of distributions,” in *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015*, 2015, pp. 294–305. 2

[21] J. Li, Y. Rabani, L. J. Schulman, and C. Swamy, “Learning arbitrary statistical mixtures of discrete distributions,” in *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015*, 2015. 2

[22] C. L. Canonne, I. Diakonikolas, T. Gouleakis, and R. Rubinfeld, “Testing shape restrictions of discrete distributions,” in *33rd Symposium on Theoretical Aspects of Computer Science, STACS 2016*, 2016, pp. 25:1–25:14. 2

[23] C. L. Canonne, “Are few bins enough: Testing histogram distributions,” in *Proceedings of the 35th ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems, PODS 2016*, 2016, pp. 455–463. 2

[24] I. Diakonikolas, D. M. Kane, and A. Stewart, “Optimal Learning via the Fourier Transform for Sums of Independent Integer Random Variables,” in *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, 2016, pp. 831–849. 2, 3, 4

[25] ——, “The fourier transform of poisson multinomial distributions and its algorithmic applications,” in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 1060–1073. 2, 3

[26] C. Daskalakis, A. De, G. Kamath, and C. Tzamos, “A size-free CLT for poisson multinomials and its applications,” in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, 2016, pp. 1074–1086. 2, 3, 6

[27] L. Chen, L. Goldstein, and Q.-M. Shao, *Normal Approximation by Stein’s Method*. Springer, 2011. 2

[28] B. Roos, “Binomial approximation to the Poisson binomial distribution: The Krawtchouk expansion,” *Theory Probab. Appl.*, vol. 45, pp. 328–344, 2000. 2

[29] I. Diakonikolas, D. M. Kane, and A. Stewart, “Properly Learning Poisson Binomial Distributions in Almost Polynomial Time,” in *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, 2016, pp. 850–878. 2, 3

[30] C. Daskalakis, G. Kamath, and C. Tzamos, “On the Structure, Covering, and Learning of Poisson Multinomial Distributions,” 2015, to appear in FOCS 2015. Available at http://arxiv.org/pdf/1504.08363v2.pdf. 2, 3, 5

[31] Y. Cheng, I. Diakonikolas, and A. Stewart, “Playing Anonymous Games Using Simple Strategies,” in *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, ser. SODA ’17, 2017, pp. 616–631. 2

[32] A. De, “Boolean Function Analysis Meets Stochastic Optimization: An Approximation Scheme for Stochastic Knapsack,” in *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*, 2018, pp. 1286–1305. 2

[33] C. Daskalakis, I. Diakonikolas, and R. Servedio, “Learning Poisson Binomial Distributions,” in *STOC*, 2012, pp. 709–728. 3, 5, 6

[34] A. De, P. M. Long, and R. Servedio, “Learning sums of independent random variables with sparse collective support,” 2018, available at https://arxiv.org/abs/1807.07013. 5, 7

[35] A. Wigderson and A. Yehudayoff, “Population Recovery and Partial Identification,” in *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS 2012, New Brunswick, NJ*, 2012, pp. 390–399. 5

[36] C. Daskalakis, I. Diakonikolas, and R. A. Servedio, “Learning *k*-Modal Distributions via Testing,” *Theory of Computing*, vol. 10, pp. 535–570, 2014. 5

[37] L. Devroye and G. Lugosi, *Combinatorial methods in density estimation*. Springer: Springer Series in Statistics, 2001. 6, 7

[38] A. De, I. Diakonikolas, and R. Servedio, “Learning from Satisfying Assignments,” in *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2015, pp. 478–497. 6

[39] A. Barbour and A. Xia, “Poisson perturbations,” *European Series in Applied and Industrial Mathematics. Probability and Statistics*, vol. 3, pp. 131–150, 1999. [Online]. Available: http://dx.doi.org/10.1051/ps:1999106 7

[40] A. Röllin, “Translated Poisson Approximation Using Exchangeable Pair Couplings,” *Annals of Applied Probability*, vol. 17, no. 5/6, pp. 1596–1614, 2007. 8

[41] A. D. Barbour, 2015, personal communication. 8, 9

[42] P. Gopalan, R. Meka, O. Reingold, and D. Zuckerman, “Pseudorandom generators for combinatorial shapes,” in *STOC*, 2011, pp. 253–262. 8

[43] T. Tao, *Higher Order Fourier Analysis*, ser. Graduate Texts in Mathematics. American Mathematical Society, 2014, no. 1. 10

[44] I. E. Shparlinski, “Distribution of modular inverses and multiples of small integers and the sato-tate conjecture on average,” *Michigan Mathematical Journal*, vol. 56, no. 1, pp. 99–111, 2008. 11