Learning Poisson Binomial Distributions

Constantinos Daskalakis*
MIT
costis@csail.mit.edu

Ilias Diakonikolas†
University of Edinburgh
ilias.d@ed.ac.uk

Rocco A. Servedio ‡
Columbia University
rocco@cs.columbia.edu

December 7, 2013

Abstract

We consider a basic problem in unsupervised learning: learning an unknown *Poisson Binomial Distribution.* A Poisson Binomial Distribution (PBD) over $\{0, 1, \ldots, n\}$ is the distribution of a sum of n independent Bernoulli random variables which may have arbitrary, potentially non-equal, expectations. These distributions were first studied by S. Poisson in 1837 [Poi37] and are a natural n-parameter generalization of the familiar Binomial Distribution. Surprisingly, prior to our work this basic learning problem was poorly understood, and known results for it were far from optimal.

We essentially settle the complexity of the learning problem for this basic class of distributions. As our first main result we give a highly efficient algorithm which learns to ϵ-accuracy (with respect to the total variation distance) using $\tilde{O}(1/\epsilon^3)$ samples *independent of* n. The running time of the algorithm is *quasilinear* in the size of its input data, i.e., $\tilde{O}(\log(n)/\epsilon^3)$ bit-operations.[1] (Observe that each draw from the distribution is a $\log(n)$-bit string.) Our second main result is a *proper* learning algorithm that learns to ϵ-accuracy using $\tilde{O}(1/\epsilon^2)$ samples, and runs in time $(1/\epsilon)^{\mathrm{poly}(\log(1/\epsilon))} \cdot \log n$. This is nearly optimal, since any algorithm for this problem must use $\Omega(1/\epsilon^2)$ samples. We also give positive and negative results for some extensions of this learning problem to weighted sums of independent Bernoulli random variables.

1 Introduction

We begin by considering a somewhat fanciful scenario: You are the manager of an independent weekly newspaper in a city of n people. Each week the i-th inhabitant of the city independently picks up a copy of your paper with probability p_i. Of course you do not know the values $p_1, \ldots, p_n$; each week you only see the total number of papers that have been picked up. For many reasons (advertising, production, revenue analysis, etc.) you would like to have a detailed "snapshot" of the probability distribution (pdf) describing how many readers you have each week. *Is there an efficient algorithm to construct a high-accuracy approximation of the pdf from a number of observations that is* independent *of the population* n*?* We show that the answer is "yes."

A *Poisson Binomial Distribution* of order n is the distribution of a sum

$$X = \sum_{i=1}^{n} X_i,$$

where $X_1, \ldots, X_n$ are independent Bernoulli (0/1) random variables. The expectations $(\mathbf{E}[X_i] = p_i)_i$ need not all be the same, and thus these distributions generalize the Binomial distribution $\mathrm{Bin}(n, p)$ and, indeed, comprise

*Research supported by NSF CAREER award CCF-0953960 and by a Sloan Foundation Fellowship.

†Research supported by a Simons Foundation Postdoctoral Fellowship. Some of this work was done while at Columbia University, supported by NSF grant CCF-0728736, and by an Alexander S. Onassis Foundation Fellowship.

‡Supported by NSF grants CNS-0716245, CCF-1115703, and CCF-1319788 and by DARPA award HR0011-08-1-0069.

[1] We write $\tilde{O}(\cdot)$ to hide factors which are polylogarithmic in the argument to $\tilde{O}(\cdot)$; thus, for example, $\tilde{O}(a \log b)$ denotes a quantity which is $O(a \log b \cdot \log^c(a \log b))$ for some absolute constant c.

a much richer class of distributions. (See Section 1.2 below.) It is believed that Poisson [Poi37] was the first to consider this extension of the Binomial distribution[2] and the distribution is sometimes referred to as "Poisson's Binomial Distribution" in his honor; we shall simply call these distributions PBDs.

PBDs are one of the most basic classes of discrete distributions; indeed, they are arguably the simplest n-parameter probability distribution that has some nontrivial structure. As such they have been intensely studied in probability and statistics (see Section 1.2) and arise in many settings; for example, we note here that tail bounds on PBDs form an important special case of Chernoff/Hoeffding bounds [Che52, Hoe63, DP09]. In application domains, PBDs have many uses in research areas such as survey sampling, case-control studies, and survival analysis, see e.g., [CL97] for a survey of the many uses of these distributions in applications. Given the simplicity and ubiquity of these distributions, it is quite surprising that the problem of *density estimation* for PBDs (i.e., learning an unknown PBD from independent samples) is not well understood in the statistics or learning theory literature. *This is the problem we consider, and essentially settle, in this paper.*

We work in a natural PAC-style model of learning an unknown discrete probability distribution which is essentially the model of [KMR$^+$94]. In this learning framework for our problem, the learner is provided with independent samples drawn from an unknown PBD X. Using these samples, the learner must with probability at least $1-\delta$ output a hypothesis distribution $\hat{X}$ such that the total variation distance $d_{\rm TV}(X, \hat{X})$ is at most ϵ, where $\epsilon, \delta > 0$ are accuracy and confidence parameters that are provided to the learner.[3] A *proper* learning algorithm in this framework outputs a distribution that is itself a Poisson Binomial Distribution, i.e., a vector $\hat{p} = (\hat{p}_1, \ldots, \hat{p}_n)$ which describes the hypothesis PBD $\hat{X} = \sum_{i=1}^n \hat{X}_i$ where $\mathbf{E}[\hat{X}_i] = \hat{p}_i$.

1.1 Our results.

Our main result is an efficient algorithm for learning PBDs from $\tilde{O}(1/\epsilon^2)$ many samples independent of $[n]$. Since PBDs are an n-parameter family of distributions over the domain $[n]$, we view such a tight bound as a surprising result. We prove:

Theorem 1 (Main Theorem). *Let $X = \sum_{i=1}^n X_i$ be an unknown PBD.*

1. **[Learning PBDs from constantly many samples]** *There is an algorithm with the following properties: given n, ϵ, δ and access to independent draws from X, the algorithm uses*

$$\tilde{O}\left((1/\epsilon^3) \cdot \log(1/\delta)\right)$$

samples from X, performs

$$\tilde{O}\left((1/\epsilon^3) \cdot \log n \cdot \log^2 \frac{1}{\delta}\right)$$

bit operations, and with probability at least $1-\delta$ outputs a (succinct description of a) distribution $\hat{X}$ over $[n]$ which is such that $d_{\rm TV}(\hat{X}, X) \le \epsilon$.

2. **[Properly learning PBDs from constantly many samples]** *There is an algorithm with the following properties: given n, ϵ, δ and access to independent draws from X, the algorithm uses*

$$\tilde{O}(1/\epsilon^2) \cdot \log(1/\delta)$$

samples from X, performs

$$(1/\epsilon)^{O\left(\log^2(1/\epsilon)\right)} \cdot \tilde{O}\left(\log n \cdot \log \frac{1}{\delta}\right)$$

bit operations, and with probability at least $1-\delta$ outputs a (succinct description of a) vector $\hat{p} = (\hat{p}_1, \ldots, \hat{p}_n)$ defining a PBD $\hat{X}$ such that $d_{\rm TV}(\hat{X}, X) \le \epsilon$.

[2] We thank Yuval Peres and Sam Watson for this information [PW11].

[3] [KMR$^+$94] used the Kullback-Leibler divergence as their distance measure but we find it more natural to use variation distance.

We note that, since every sample drawn from X is a $\log(n)$-bit string, for constant δ the number of bit-operations performed by our first algorithm is *quasilinear* in the length of its input. Moreover, the sample complexity of both algorithms is close to optimal, since $\Omega(1/\epsilon^2)$ samples are required even to distinguish the (simpler) Binomial distributions $\mathrm{Bin}(n, 1/2)$ and $\mathrm{Bin}(n, 1/2+\epsilon/\sqrt{n})$, which have total variation distance $\Omega(\epsilon)$. Indeed, in view of this observation, our second algorithm is essentially sample-optimal.

Motivated by these strong learning results for PBDs, we also consider learning a more general class of distributions, namely distributions of the form $X = \sum_{i=1}^n w_i X_i$ which are *weighted* sums of independent Bernoulli random variables. We give an algorithm which uses $O(\log n)$ samples and runs in $\mathrm{poly}(n)$ time if there are only constantly many different weights in the sum:

Theorem 2 (Learning sums of weighted independent Bernoulli random variables). *Let $X = \sum_{i=1}^n a_i X_i$ be a weighted sum of unknown independent Bernoullis such that there are at most k different values among $a_1, \dots, a_n$. Then there is an algorithm with the following properties: given n, ϵ, δ, $a_1, \dots, a_n$ and access to independent draws from X, it uses*

$$\widetilde{O}(k/\epsilon^2) \cdot \log(n) \cdot \log(1/\delta)$$

samples from X, runs in time

$$\mathrm{poly}\left(n^k \cdot \epsilon^{-k \log^2(1/\epsilon)}\right) \cdot \log(1/\delta),$$

and with probability at least $1-\delta$ outputs a hypothesis vector $\hat{p} \in [0,1]^n$ defining independent Bernoulli random variables $\hat{X}_i$ with $\mathbf{E}[\hat{X}_i] = \hat{p}_i$ such that $d_{\mathrm{TV}}(\hat{X}, X) \leq \epsilon$, where $\hat{X} = \sum_{i=1}^n a_i \hat{X}_i$.

To complement Theorem 2, we also show that if there are many distinct weights in the sum, then even for weights with a very simple structure any learning algorithm must use many samples:

Theorem 3 (Sample complexity lower bound for learning sums of weighted independent Bernoullis). *Let $X = \sum_{i=1}^n i \cdot X_i$ be a weighted sum of unknown independent Bernoullis (where the i-th weight is simply i). Let L be any learning algorithm which, given n and access to independent draws from X, outputs a hypothesis distribution $\hat{X}$ such that $d_{\mathrm{TV}}(\hat{X}, X) \leq 1/25$ with probability at least $e^{-o(n)}$. Then L must use $\Omega(n)$ samples.*

1.2 Related work.

At a high level, there has been a recent surge of interest in the theoretical computer science community on fundamental algorithmic problems involving basic types of probability distributions, see e.g., [KMV10, MV10, BS10, VV11] and other recent papers; our work may be considered as an extension of this theme. More specifically, there is a broad literature in probability theory studying various properties of PBDs; see [Wan93] for an accessible introduction to some of this work. In particular, many results study approximations to the Poisson Binomial distribution via simpler distributions. In a well-known result, Le Cam [Cam60] shows that for any PBD $X = \sum_{i=1}^n X_i$ with $\mathbf{E}[X_i] = p_i$, it holds that

$$d_{\mathrm{TV}}\left(X, \mathrm{Poi}\big(\sum_{i=1}^n p_i\big)\right) \leq 2 \sum_{i=1}^n p_i^2,$$

where $\mathrm{Poi}(\lambda)$ is the Poisson distribution with parameter λ. Subsequently many other proofs of this result and similar ones were given using a range of different techniques; [HC60, Che74, DP86, BHJ92] is a sampling of work along these lines, and Steele [Ste94] gives an extensive list of relevant references. Much work has also been done on approximating PBDs by normal distributions (see e.g., [Ber41, Ess42, Mik93, Vol95]) and by Binomial distributions (see e.g., [Ehm91, Soo96, Roo00]). These results provide structural information about PBDs that can be well-approximated via simpler distributions, but fall short of our goal of obtaining approximations of an unknown PBD up to *arbitrary accuracy*. Indeed, the approximations obtained in the probability literature (such as the Poisson, Normal and Binomial approximations) typically depend only on the first few moments of the target PBD, while higher moments are crucial for arbitrary approximation [Roo00].

Taking a different perspective, it is easy to show (see Section 2 of [KG71]) that every PBD is a unimodal distribution over $[n]$. The learnability of general unimodal distributions over $[n]$ is well understood: Birgé [Bir87a, Bir97] has given a computationally efficient algorithm that can learn any unimodal distribution over $[n]$ to variation distance ϵ from $O(\log(n)/\epsilon^3)$ samples, and has shown that any algorithm must use $\Omega(\log(n)/\epsilon^3)$ samples. (The [Bir87a, Bir97] upper and lower bounds are stated for continuous unimodal distributions, but the arguments are easily adapted to the discrete case.) Our main result, Theorem 1, shows that the additional PBD assumption can be leveraged to obtain sample complexity *independent of* n with a computationally highly efficient algorithm.

So, how might one leverage the structure of PBDs to remove n from the sample complexity? A first observation is that a PBD assigns $1-\epsilon$ of its mass to $O_\epsilon(\sqrt{n})$ points. So one could draw samples to (approximately) identify these points and then try to estimate the probability assigned to each such point, but clearly such an approach, if followed naïvely, would give $\mathrm{poly}(n)$ sample complexity. Alternatively, one could run Birgé's algorithm on the restricted support of size $O_\epsilon(\sqrt{n})$, but that will not improve the asymptotic sample complexity. A different approach would be to construct a small ϵ-cover (under the total variation distance) of the space of all PBDs on n variables. Indeed, if such a cover has size N, it can be shown (see Lemma 10 in Section 3.1, or Chapter 7 of [DL01])) that a target PBD can be learned from $O(\log(N)/\epsilon^2)$ samples. Still it is easy to argue that any cover needs to have size $\Omega(n)$, so this approach too gives a $\log(n)$ dependence in the sample complexity.

Our approach, which removes n completely from the sample complexity, requires a refined understanding of the structure of the set of all PBDs on n variables, in fact one that is more refined than the understanding provided by the aforementioned results (approximating a PBD by a Poisson, Normal, or Binomial distribution). We give an outline of the approach in the next section.

1.3 Our approach.

The starting point of our algorithm for learning PBDs is a theorem of [DP11, Das08] that gives detailed information about the structure of a small ϵ-cover (under the total variation distance) of the space of all PBDs on n variables (see Theorem 4). Roughly speaking, this result says that every PBD is either close to a PBD whose support is sparse, or is close to a translated "heavy" Binomial distribution. Our learning algorithm exploits this structure of the cover; it has two subroutines corresponding to these two different types of distributions that the cover contains. First, assuming that the target PBD is close to a sparsely supported distribution, it runs Birgé's unimodal distribution learner over a carefully selected subinterval of $[n]$ to construct a hypothesis H_S; the (purported) sparsity of the distribution makes it possible for this algorithm to use $\tilde{O}(1/\epsilon^3)$ samples independent of n. Then, assuming that the target PBD is close to a translated "heavy" Binomial distribution, the algorithm constructs a hypothesis Translated Poisson Distribution H_P [RÖ7] whose mean and variance match the estimated mean and variance of the target PBD; we show that H_P is close to the target PBD if the target PBD is not close to any sparse distribution in the cover. At this point the algorithm has two hypothesis distributions, H_S and H_P, one of which should be good; it remains to select one as the final output hypothesis. This is achieved using a form of "hypothesis testing" for probability distributions.

The above sketch captures the main ingredients of Part (1) of Theorem 1, but additional work needs to be done to get the proper learning algorithm of Part (2). For the non-sparse case, first note that the Translated Poisson hypothesis H_P is not a PBD. Via a sequence of transformations we are able to show that the Translated Poisson hypothesis H_P can be converted to a Binomial distribution $\mathrm{Bin}(n', p)$ for some $n' \leq n$. To handle the sparse case, we use an alternate learning approach: instead of using Birgé's unimodal algorithm (which would incur a sample complexity of $\Omega(1/\epsilon^3)$), we first show that, in this case, there exists an efficiently constructible $O(\epsilon)$-cover of size $(1/\epsilon)^{O(\log^2(1/\epsilon))}$, and then apply a general learning result that we now describe.

The general learning result that we use (Lemma 10) is the following: We show that for any class $\mathcal{S}$ of target distributions, if $\mathcal{S}$ has an ϵ-cover of size N then there is a generic algorithm for learning an unknown distribution from $\mathcal{S}$ to accuracy $O(\epsilon)$ that uses $O((\log N)/\epsilon^2)$ samples. Our approach is rather similar to the algorithm of [DL01] for choosing a density estimate (but different in some details); it works by carrying out a tournament that matches every pair of distributions in the cover against each other. Our analysis shows that with high probability some ϵ-accurate distribution in the cover will survive the tournament undefeated, and that any undefeated distribution will with high probability be $O(\epsilon)$-accurate.

Applying this general result to the $O(\epsilon)$-cover of size $(1/\epsilon)^{O(\log^2(1/\epsilon))}$ described above, we obtain a PBD that is $O(\epsilon)$-close to the target (this accounts for the increased running time in Part (2) versus Part (1)). We stress that for both the non-proper and proper learning algorithms sketched above, many technical subtleties and challenges arise in implementing the high-level plan given above, requiring a careful and detailed analysis.

We prove Theorem 2 using the general approach of Lemma 10 specialized to weighted sums of independent Bernoullis with constantly many distinct weights. We show how the tournament can be implemented efficiently for the class $\mathcal{S}$ of weighted sums of independent Bernoullis with constantly many distinct weights, and thus obtain Theorem 2. Finally, the lower bound of Theorem 3 is proved by a direct information-theoretic argument.

1.4 Preliminaries.

Distributions. For a distribution X supported on $[n] = \{0, 1, \ldots, n\}$ we write $X(i)$ to denote the value $\Pr[X = i]$ of the probability density function (pdf) at point i, and $X(\leq i)$ to denote the value $\Pr[X \leq i]$ of the cumulative density function (cdf) at point i. For $S \subseteq [n]$, we write $X(S)$ to denote $\sum_{i \in S} X(i)$ and X_S to denote the conditional distribution of X restricted to S. Sometimes we write $X(I)$ and X_I for a subset $I \subseteq [0, n]$, meaning $X(I \cap [n])$ and $X_{I \cap [n]}$ respectively.

Total Variation Distance. Recall that the *total variation distance* between two distributions X and Y over a finite domain D is

$$d_{\mathrm{TV}}\left(X, Y\right) \quad := \quad (1/2) \cdot \sum_{\alpha \in D} |X(\alpha) - Y(\alpha)| = \max_{S \subseteq D} [X(S) - Y(S)].$$

Similarly, if X and Y are two random variables ranging over a finite set, their total variation distance $d_{\mathrm{TV}}(X, Y)$ is defined as the total variation distance between their distributions. For convenience, we will often blur the distinction between a random variable and its distribution.

Covers. Fix a finite domain D, and let $\mathcal{P}$ denote some set of distributions over D. Given $\delta > 0$, a subset $\mathcal{Q} \subseteq \mathcal{P}$ is said to be a *δ-cover of $\mathcal{P}$* (w.r.t. the total variation distance) if for every distribution P in $\mathcal{P}$ there exists some distribution Q in $\mathcal{Q}$ such that $d_{\mathrm{TV}}(P, Q) \leq \delta$. We sometimes say that distributions P, Q are *δ-neighbors* if $d_{\mathrm{TV}}(P, Q) \leq \delta$. If this holds, we also say that P is δ-close to Q and vice versa.

Poisson Binomial Distribution. A *Poisson binomial distribution of order $n \in \mathbb{N}$* is the discrete probability distribution of the sum $\sum_{i=1}^{n} X_i$ of n mutually independent Bernoulli random variables $X_1, \ldots, X_n$. We denote the set of all Poisson binomial distributions of order n by $\mathcal{S}_n$ and, if n is clear from context, just $\mathcal{S}$.

A Poisson binomial distribution $D \in \mathcal{S}_n$ can be represented uniquely as a vector $(p_i)_{i=1}^{n}$ satisfying $0 \leq p_1 \leq p_2 \leq \ldots \leq p_n \leq 1$. To go from $D \in \mathcal{S}_n$ to its corresponding vector, we find a collection $X_1, \ldots, X_n$ of mutually independent Bernoullis such that $\sum_{i=1}^{n} X_i$ is distributed according to D and $\mathbf{E}[X_1] \leq \ldots \leq \mathbf{E}[X_n]$. (Such a collection exists by the definition of a Poisson binomial distribution.) Then we set $p_i = \mathbf{E}[X_i]$ for all i. Lemma 1 of [DP13] shows that the resulting vector $(p_1, \ldots, p_n)$ is unique.

We denote by $\mathrm{PBD}(p_1, \ldots, p_n)$ the distribution of the sum $\sum_{i=1}^{n} X_i$ of mutually independent indicators $X_1, \ldots, X_n$ with expectations $p_i = \mathbf{E}[X_i]$, for all i. Given the above discussion $\mathrm{PBD}(p_1, \ldots, p_n)$ is unique up to permutation of the p_i's. We also sometimes write $\{X_i\}$ to denote the distribution of $\sum_{i=1}^{n} X_i$. Note the difference between $\{X_i\}$, which refers to the distribution of $\sum_i X_i$, and $\{X_i\}_i$, which refers to the underlying collection of mutually independent Bernoulli random variables.

Translated Poisson Distribution. We will make use of the translated Poisson distribution for approximating the Poisson Binomial distribution. We define the translated Poisson distribution, and state a known result on how well it approximates the Poisson Binomial distribution.

Definition 1 ([RÖ7]). *We say that an integer random variable Y is distributed according to the* translated Poisson distribution with parameters μ and σ^2 *iff Y can be written as*

$$Y = \lfloor \mu - \sigma^2 \rfloor + Z,$$

where $\{\mu - \sigma^2\}$ represents the fractional part of $\mu - \sigma^2$, and Z is a random variable distributed according to $\text{Poisson}(\sigma^2 + \{\mu - \sigma^2\})$.

Lemma 1 (see (3.4) of [RÖ7]). *Let $J_1, \ldots, J_n$ be independent random indicators with $\mathbf{E}[J_i] = p_i$. Then*

$$d_{\text{TV}}\left(\sum_{i=1}^{n} J_i, TP(\mu, \sigma^2)\right) \leq \frac{\sqrt{\sum_{i=1}^{n} p_i^3(1-p_i)} + 2}{\sum_{i=1}^{n} p_i(1-p_i)},$$

where $\mu = \sum_{i=1}^{n} p_i$ and $\sigma^2 = \sum_{i=1}^{n} p_i(1-p_i)$.

The following bound on the total variation distance between translated Poisson distributions will be useful.

Lemma 2 (Lemma 2.1 of [BL06]). *For $\mu_1, \mu_2 \in \mathbb{R}$ and $\sigma_1^2, \sigma_2^2 \in \mathbb{R}_+$ with $\lfloor \mu_1 - \sigma_1^2 \rfloor \leq \lfloor \mu_2 - \sigma_2^2 \rfloor$, we have*

$$d_{\text{TV}}(TP(\mu_1, \sigma_1^2), TP(\mu_2, \sigma_2^2)) \leq \frac{|\mu_1 - \mu_2|}{\sigma_1} + \frac{|\sigma_1^2 - \sigma_2^2| + 1}{\sigma_1^2}.$$

Running Times, and Bit Complexity. Throughout this paper, we measure the running times of our algorithms in numbers of bit operations. For a positive integer n, we denote by $\langle n \rangle$ its description complexity in binary, namely $\langle n \rangle = \lceil \log_2 n \rceil$. Moreover, we represent a positive rational number q as $\frac{q_1}{q_2}$, where q_1 and q_2 are relatively prime positive integers. The description complexity of q is defined to be $\langle q \rangle = \langle q_1 \rangle + \langle q_2 \rangle$. We will assume that all ϵ's and δ's input to our algorithms are rational numbers.

2 Learning an unknown sum of Bernoulli random variables from $\text{poly}(1/\epsilon)$ samples

In this section, we prove Theorem 1 by providing a sample- and time-efficient algorithm for learning an unknown PBD $X = \sum_{i=1}^{n} X_i$. We start with an important ingredient in our analysis.

A cover for PBDs. We make use of the following theorem, which provides a cover of the set $\mathcal{S} = \mathcal{S}_n$ of all PBDs of order-n. The theorem was given implicitly in [DP11] and explicitly as Theorem 1 in [DP13].

Theorem 4 (Cover for PBDs). *For all $\epsilon > 0$, there exists an ϵ-cover $\mathcal{S}_\epsilon \subseteq \mathcal{S}$ of $\mathcal{S}$ such that*

1. *$|\mathcal{S}_\epsilon| \leq n^2 + n \cdot \left(\frac{1}{\epsilon}\right)^{O(\log^2 1/\epsilon)}$; and*

2. *$\mathcal{S}_\epsilon$ can be constructed in time linear in its representation size, i.e., $O(n^2 \log n) + O(n \log n) \cdot \left(\frac{1}{\epsilon}\right)^{O(\log^2 1/\epsilon)}$.*

Moreover, if $\{Y_i\} \in \mathcal{S}_\epsilon$, then the collection of random variables $\{Y_i\}_i$ has one of the following forms, where $k = k(\epsilon) \leq C/\epsilon$ is a positive integer, for some absolute constant $C > 0$:

(i) (k-Sparse Form) There is some $\ell \leq k^3 = O(1/\epsilon^3)$ such that, for all $i \leq \ell$, $\mathbf{E}[Y_i] \in \left\{\frac{1}{k^2}, \frac{2}{k^2}, \ldots, \frac{k^2-1}{k^2}\right\}$ and, for all $i > \ell$, $\mathbf{E}[Y_i] \in \{0, 1\}$.

(ii) *(k-heavy Binomial Form) There is some $\ell \in \{1, \dots, n\}$ and $q \in \left\{\frac{1}{n}, \frac{2}{n}, \dots, \frac{n}{n}\right\}$ such that, for all $i \le \ell$, $\mathbf{E}[Y_i] = q$ and, for all $i > \ell$, $\mathbf{E}[Y_i] = 0$; moreover, ℓ, q satisfy $\ell q \ge k^2$ and $\ell q(1-q) \ge k^2 - k - 1$.*

Finally, for every $\{X_i\} \in \mathcal{S}$ for which there is no ϵ-neighbor in $\mathcal{S}_\epsilon$ that is in sparse form, there exists some $\{Y_i\} \in \mathcal{S}_\epsilon$ in k-heavy Binomial form such that

(iii) *$d_{\mathrm{TV}}(\sum_i X_i, \sum_i Y_i) \le \epsilon$; and*

(iv) *if $\mu = \mathbf{E}[\sum_i X_i]$, $\mu' = \mathbf{E}[\sum_i Y_i]$, $\sigma^2 = \mathrm{Var}[\sum_i X_i]$ and $\sigma'^2 = \mathrm{Var}[\sum_i Y_i]$, then $|\mu - \mu'| = O(1)$ and $|\sigma^2 - \sigma'^2| = O(1 + \epsilon \cdot (1 + \sigma^2))$.*

We remark that the cover theorem as stated in [DP13] does not include the part of the above statement following "finally." We provide a proof of this extension in Appendix A.

The Basic Learning Algorithm. Theorem 1 is established by making use of algorithm `Learn-PBD` of Figure 1, with appropriate modifications. These modifications are of technical nature, and are postponed to Section 2.4.

```
Learn-PBD
  1. Run Learn-Sparse^X$(n, \epsilon, \delta/3)$ to get hypothesis distribution $H_S$.
  2. Run Learn-Poisson^X$(n, \epsilon, \delta/3)$ to get hypothesis distribution $H_P$.
  3. Return the distribution which is the output of Choose-Hypothesis^X$(H_S, H_P, \epsilon, \delta/3)$.
```

Figure 1: `Learn-PBD`

At a high level, the subroutine `Learn-Sparse` is given sample access to X and is designed to find an ϵ-accurate hypothesis H_S with probability at least $1 - \delta/3$, if the unknown PBD X is ϵ-close to some sparse form PBD inside the cover $\mathcal{S}_\epsilon$. Similarly, `Learn-Poisson` is designed to find an ϵ-accurate hypothesis H_P, if X is not ϵ-close to a sparse form PBD (in this case, Theorem 4 implies that X must be ϵ-close to some $k(\epsilon)$-heavy Binomial form PBD). Finally, `Choose-Hypothesis` is designed to choose one of the two hypotheses H_S, H_P as being ϵ-close to X. The following subsections specify these subroutines, as well as how the algorithm can be used to establish Theorem 1. We note that `Learn-Sparse` and `Learn-Poisson` do not return the distributions H_S and H_P as a list of probabilities for every point in $[n]$. They return instead a succinct description of these distributions in order to keep the running time of the algorithm logarithmic in n. Similarly, `Choose-Hypothesis` operates with succinct descriptions of these distributions.

2.1 Learning when X is close to a sparse form PBD.

Our starting point here is the simple observation that any PBD is a unimodal distribution over the domain $\{0, 1, \dots, n\}$. (There is a simple inductive proof of this, or see Section 2 of [KG71].) This enables us to use the algorithm of Birgé [Bir97] for learning unimodal distributions. We recall Birgé's result, and refer the reader to Appendix B for an explanation of how Theorem 5 as stated below follows from [Bir97].

Theorem 5 ([Bir97])**.** *For all $n, \epsilon, \delta > 0$, there is an algorithm that draws*

$$O\left(\frac{\log n}{\epsilon^3} \log \frac{1}{\delta} + \frac{1}{\epsilon^2} \log \frac{1}{\delta} \log\log \frac{1}{\delta}\right)$$

samples from an unknown unimodal distribution X over $[n]$, does

$$\tilde{O}\left(\frac{\log^2 n}{\epsilon^3} \log^2 \frac{1}{\delta}\right)$$

bit-operations, and outputs a (succinct description of a) hypothesis distribution H over $[n]$ that has the following form: H is uniform over subintervals $[a_1, b_1], [a_2, b_2], \ldots, [a_k, b_k]$, whose union $\cup_{i=1}^{k}[a_i, b_i] = [n]$, where $k = O\left(\frac{\log n}{\epsilon}\right)$. In particular, the algorithm outputs the lists a_1 through a_k and b_1 through b_k, as well as the total probability mass that H assigns to each subinterval $[a_i, b_i]$, $i = 1, \ldots, k$. Finally, with probability at least $1 - \delta$, $d_{\mathrm{TV}}(X, H) \leq \epsilon$.

The main result of this subsection is the following:

Lemma 3. *For all $n, \epsilon', \delta' > 0$, there is an algorithm* $\texttt{Learn-Sparse}^X(n, \epsilon', \delta')$ *that draws*

$$O\left(\frac{1}{\epsilon'^3} \log \frac{1}{\epsilon'} \log \frac{1}{\delta'} + \frac{1}{\epsilon'^2} \log \frac{1}{\delta'} \log \log \frac{1}{\delta'}\right)$$

samples from a target PBD X over $[n]$, does

$$\log n \cdot \tilde{O}\left(\frac{1}{\epsilon'^3} \log^2 \frac{1}{\delta'}\right)$$

bit operations, and outputs a (succinct description of a) hypothesis distribution H_S over $[n]$ that has the following form: its support is contained in an explicitly specified interval $[a, b] \subset [n]$, where $|b - a| = O(1/\epsilon'^3)$, and for every point in $[a, b]$ the algorithm explicitly specifies the probability assigned to that point by H_S. [4] *The algorithm has the following guarantee: if X is ϵ'-close to some sparse form PBD Y in the cover $\mathcal{S}_{\epsilon'}$ of Theorem 4, then with probability at least $1 - \delta'$, $d_{\mathrm{TV}}(X, H_S) \leq c_1 \epsilon'$, for some absolute constant $c_1 \geq 1$, and the support of H_S lies in the support of Y.*

The high-level idea of Lemma 3 is quite simple. We truncate $O(\epsilon')$ of the probability mass from each end of X to obtain a conditional distribution $X_{[\hat{a},\hat{b}]}$; since X is unimodal so is $X_{[\hat{a},\hat{b}]}$. If $\hat{b} - \hat{a}$ is larger than $O(1/\epsilon'^3)$ then the algorithm outputs "fail" (and X could not have been close to a sparse-form distribution in the cover). Otherwise, we use Birgé's algorithm to learn the unimodal distribution $X_{[\hat{a},\hat{b}]}$.

Proof of Lemma 3: The Algorithm $\texttt{Learn-Sparse}^X(n, \epsilon', \delta')$ works as follows: It first draws $M = 32 \log(8/\delta')/\epsilon'^2$ samples from X and sorts them to obtain a list of values $0 \leq s_1 \leq \cdots \leq s_M \leq n$. In terms of these samples, let us define $\hat{a} := s_{\lceil 2\epsilon' M \rceil}$ and $\hat{b} := s_{\lfloor (1-2\epsilon')M \rfloor}$. We claim the following:

Claim 4. *With probability at least $1 - \delta'/2$, we have $X(\leq \hat{a}) \in [3\epsilon'/2, 5\epsilon'/2]$ and $X(\leq \hat{b}) \in [1 - 5\epsilon'/2, 1 - 3\epsilon'/2]$.*

Proof. We only show that $X(\leq \hat{a}) \geq 3\epsilon'/2$ with probability at least $1 - \delta'/8$, since the arguments for $X(\leq \hat{a}) \leq 5\epsilon'/2$, $X(\leq \hat{b}) \leq 1 - 3\epsilon'/2$ and $X(\leq \hat{b}) \geq 1 - 5\epsilon'/2$ are identical. Given that each of these conditions is met with probability at least $1 - \delta'/8$, the union bound establishes our claim.

To show that $X(\leq \hat{a}) \geq 3\epsilon'/2$ is satisfied with probability at least $1 - \delta'/8$ we argue as follows: Let $\alpha' = \max\{i \mid X(\leq i) < 3\epsilon'/2\}$. Clearly, $X(\leq \alpha') < 3\epsilon'/2$ while $X(\leq \alpha' + 1) \geq 3\epsilon'/2$. Given this, if M samples are drawn from X then the expected number of them that are $\leq \alpha'$ is at most $3\epsilon' M/2$. It follows then from the Chernoff bound that the probability that more than $\frac{7}{4}\epsilon' M$ samples are $\leq \alpha'$ is at most $e^{-(\epsilon'/4)^2 M/2} \leq \delta'/8$. Hence except with this failure probability, we have $\hat{a} \geq \alpha' + 1$, which implies that $X(\leq \hat{a}) \geq 3\epsilon'/2$. □

If $\hat{b} - \hat{a} > (C/\epsilon')^3$, where C is the constant in the statement of Theorem 4, the algorithm outputs "fail", returning the trivial hypothesis which puts probability mass 1 on the point 0. Otherwise, the algorithm runs Birgé's unimodal distribution learner (Theorem 5) on the conditional distribution $X_{[\hat{a},\hat{b}]}$, and outputs the result of Birgé's algorithm. Since X is unimodal, it follows that $X_{[\hat{a},\hat{b}]}$ is also unimodal, hence Birgé's algorithm is

[4] In particular, our algorithm will output a list of pointers, mapping every point in $[a, b]$ to some memory location where the probability assigned to that point by H_S is written.

appropriate for learning it. The way we apply Birgé's algorithm to learn $X_{[\hat{a},\hat{b}]}$ given samples from the original distribution X is the obvious one: we draw samples from X, ignoring all samples that fall outside of $[\hat{a},\hat{b}]$, until the right $O(\log(1/\delta')\log(1/\epsilon')/\epsilon'^3)$ number of samples fall inside $[\hat{a},\hat{b}]$, as required by Birgé's algorithm for learning a distribution of support of size $(C/\epsilon')^3$ with probability at least $1-\delta'/4$. Once we have the right number of samples in $[\hat{a},\hat{b}]$, we run Birgé's algorithm to learn the conditional distribution $X_{[\hat{a},\hat{b}]}$. Note that the number of samples we need to draw from X until the right $O(\log(1/\delta')\log(1/\epsilon')/\epsilon'^3)$ number of samples fall inside $[\hat{a},\hat{b}]$ is still $O(\log(1/\delta')\log(1/\epsilon')/\epsilon'^3)$, with probability at least $1-\delta'/4$. Indeed, since $X([\hat{a},\hat{b}])=1-O(\epsilon')$, it follows from the Chernoff bound that with probability at least $1-\delta'/4$, if $K=\Theta(\log(1/\delta')\log(1/\epsilon')/\epsilon'^3)$ samples are drawn from X, at least $K(1-O(\epsilon'))$ fall inside $[\hat{a},\hat{b}]$.

Analysis: It is easy to see that the sample complexity of our algorithm is as promised. For the running time, notice that, if Birgé's algorithm is invoked, it will return two lists of numbers a_1 through a_k and b_1 through b_k, as well as a list of probability masses $q_1,\dots,q_k$ assigned to each subinterval $[a_i,b_i]$, $i=1,\dots,k$, by the hypothesis distribution H_S, where $k=O(\log(1/\epsilon')/\epsilon')$. In linear time, we can compute a list of probabilities $\hat{q}_1,\dots,\hat{q}_k$, representing the probability assigned by H_S to every point of subinterval $[a_i,b_i]$, for $i=1,\dots,k$. So we can represent our output hypothesis H_S via a data structure that maintains $O(1/\epsilon'^3)$ pointers, having one pointer per point inside $[a,b]$. The pointers map points to probabilities assigned by H_S to these points. Thus turning the output of Birgé's algorithm into an explicit distribution over $[a,b]$ incurs linear overhead in our running time, and hence the running time of our algorithm is also as promised. Moreover, we also note that the output distribution has the promised structure, since in one case it has a single atom at 0 and in the other case it is the output of Birgé's algorithm on a distribution of support of size $(C/\epsilon')^3$.

It only remains to justify the last part of the lemma. Let Y be the sparse-form PBD that X is close to; say that Y is supported on $\{a',\dots,b'\}$ where $b'-a'\le(C/\epsilon')^3$. Since X is ϵ'-close to Y in total variation distance it must be the case that $X(\le a'-1)\le\epsilon'$. Since $X(\le\hat{a})\ge 3\epsilon'/2$ by Claim 4, it must be the case that $\hat{a}\ge a'$. Similar arguments give that $\hat{b}\le b'$. So the interval $[\hat{a},\hat{b}]$ is contained in $[a',b']$ and has length at most $(C/\epsilon')^3$. This means that Birgé's algorithm is indeed used correctly by our algorithm to learn $X_{[\hat{a},\hat{b}]}$, with probability at least $1-\delta'/2$ (that is, unless Claim 4 fails). Now it follows from the correctness of Birgé's algorithm (Theorem 5) and the discussion above, that the hypothesis H_S output when Birgé's algorithm is invoked satisfies $d_{\mathrm{TV}}(H_S,X_{[\hat{a},\hat{b}]})\le\epsilon'$, with probability at least $1-\delta'/2$, i.e., unless either Birgé's algorithm fails, or we fail to get the right number of samples landing inside $[\hat{a},\hat{b}]$. To conclude the proof of the lemma we note that:

$$\begin{aligned}
2d_{\mathrm{TV}}(X,X_{[\hat{a},\hat{b}]}) &= \sum_{i\in[\hat{a},\hat{b}]}|X_{[\hat{a},\hat{b}]}(i)-X(i)|+\sum_{i\notin[\hat{a},\hat{b}]}|X_{[\hat{a},\hat{b}]}(i)-X(i)|\\
&= \sum_{i\in[\hat{a},\hat{b}]}\left|\frac{1}{X([\hat{a},\hat{b}])}X(i)-X(i)\right|+\sum_{i\notin[\hat{a},\hat{b}]}X(i)\\
&= \sum_{i\in[\hat{a},\hat{b}]}\left|\frac{1}{1-O(\epsilon')}X(i)-X(i)\right|+O(\epsilon')\\
&= \frac{O(\epsilon')}{1-O(\epsilon')}\sum_{i\in[\hat{a},\hat{b}]}\left|X(i)\right|+O(\epsilon')\\
&= O(\epsilon').
\end{aligned}$$

So the triangle inequality gives: $d_{\mathrm{TV}}(H_S,X)=O(\epsilon')$, and Lemma 3 is proved. □

2.2 Learning when X is close to a k-heavy Binomial Form PBD.

Lemma 5. *For all $n,\epsilon',\delta'>0$, there is an algorithm* `Learn-Poisson`$^X(n,\epsilon',\delta')$ *that draws*

$$O(\log(1/\delta')/\epsilon'^2)$$

samples from a target PBD X over $[n]$, does

$$O(\log n \cdot \log(1/\delta')/\epsilon'^2)$$

bit operations, and returns two parameters $\hat{\mu}$ and $\hat{\sigma}^2$. The algorithm has the following guarantee: Suppose X is not ϵ'-close to any sparse form PBD in the cover $\mathcal{S}_{\epsilon'}$ of Theorem 4. Let $H_P = TP(\hat{\mu}, \hat{\sigma}^2)$ be the translated Poisson distribution with parameters $\hat{\mu}$ and $\hat{\sigma}^2$. Then with probability at least $1-\delta'$ we have $d_{\rm TV}(X, H_P) \le c_2\epsilon'$ for some absolute constant $c_2 \ge 1$.

Our proof plan is to exploit the structure of the cover of Theorem 4. In particular, if X is not ϵ'-close to any sparse form PBD in the cover, it must be ϵ'-close to a PBD in heavy Binomial form with approximately the same mean and variance as X, as specified by the final part of the cover theorem. Hence, a natural strategy is to obtain estimates $\hat{\mu}$ and $\hat{\sigma}^2$ of the mean and variance of the unknown PBD X, and output as a hypothesis a translated Poisson distribution with parameters $\hat{\mu}$ and $\hat{\sigma}^2$. We show that this strategy is a successful one. Before providing the details, we highlight two facts that we will establish in the subsequent analysis and that will be used later. The first is that, assuming X is not ϵ'-close to any sparse form PBD in the cover $\mathcal{S}_{\epsilon'}$, its variance σ^2 satisfies

$$\sigma^2 = \Omega(1/\epsilon'^2) \ge \theta^2 \quad \text{for some universal constant } \theta. \tag{1}$$

The second is that under the same assumption, the estimates $\hat{\mu}$ and $\hat{\sigma}^2$ of the mean μ and variance σ^2 of X that we obtain satisfy the following bounds with probability at least $1-\delta$:

$$|\mu - \hat{\mu}| \le \epsilon' \cdot \sigma \quad \text{and} \quad |\sigma^2 - \hat{\sigma}^2| \le \epsilon' \cdot \sigma^2. \tag{2}$$

Proof of Lemma 5: We start by showing that we can estimate the mean and variance of the target PBD X.

Lemma 6. *For all $n, \epsilon, \delta > 0$, there exists an algorithm $\mathcal{A}(n, \epsilon, \delta)$ with the following properties: given access to a PBD X of order n, it produces estimates $\hat{\mu}$ and $\hat{\sigma}^2$ for $\mu = \mathbf{E}[X]$ and $\sigma^2 = \mathrm{Var}[X]$ respectively such that with probability at least $1-\delta$:*

$$|\mu - \hat{\mu}| \le \epsilon \cdot \sigma \qquad \textit{and} \qquad |\sigma^2 - \hat{\sigma}^2| \le \epsilon \cdot \sigma^2 \sqrt{4 + \frac{1}{\sigma^2}}.$$

The algorithm uses

$$O(\log(1/\delta)/\epsilon^2)$$

samples and runs in time

$$O(\log n \log(1/\delta)/\epsilon^2).$$

Proof. We treat the estimation of μ and σ^2 separately. For both estimation problems we show how to use $O(1/\epsilon^2)$ samples to obtain estimates $\hat{\mu}$ and $\hat{\sigma}^2$ achieving the required guarantees with probability at least $2/3$. Then a routine procedure allows us to boost the success probability to $1-\delta$ at the expense of a multiplicative factor $O(\log 1/\delta)$ on the number of samples. While we omit the details of the routine boosting argument, we remind the reader that it involves running the weak estimator $O(\log 1/\delta)$ times to obtain estimates $\hat{\mu}_1, \ldots, \hat{\mu}_{O(\log 1/\delta)}$ and outputting the median of these estimates, and similarly for estimating σ^2.

We proceed to specify and analyze the weak estimators for μ and σ^2 separately:

- *Weak estimator for μ:* Let $Z_1, \ldots, Z_m$ be independent samples from X, and let $\hat{\mu} = \frac{\sum_i Z_i}{m}$. Then

  $$\mathbf{E}[\hat{\mu}] = \mu \ \text{ and } \ \mathrm{Var}[\hat{\mu}] = \frac{1}{m}\mathrm{Var}[X] = \frac{1}{m}\sigma^2.$$

  So Chebyshev's inequality implies that

  $$\Pr[|\hat{\mu} - \mu| \ge t\sigma/\sqrt{m}] \le \frac{1}{t^2}.$$

  Choosing $t = \sqrt{3}$ and $m = \lceil 3/\epsilon^2 \rceil$, the above imply that $|\hat{\mu} - \mu| \le \epsilon\sigma$ with probability at least $2/3$.

- *Weak estimator for σ^2:* Let $Z_1, \ldots, Z_m$ be independent samples from X, and let $\hat{\sigma}^2 = \frac{\sum_i (Z_i - \frac{1}{m}\sum_i Z_i)^2}{m-1}$ be the unbiased sample variance. (Note the use of Bessel's correction.) Then it can be checked [Joh03] that

  $$\mathbf{E}[\hat{\sigma}^2] = \sigma^2 \text{ and } \mathrm{Var}[\hat{\sigma}^2] = \sigma^4 \left(\frac{2}{m-1} + \frac{\kappa}{m} \right),$$

  where κ is the kurtosis of the distribution of X. To bound κ in terms of σ^2 suppose that $X = \sum_{i=1}^n X_i$, where $\mathbf{E}[X_i] = p_i$ for all i. Then

  $$\begin{aligned} \kappa &= \frac{1}{\sigma^4} \sum_i (1 - 6p_i(1-p_i))(1-p_i)p_i \qquad \text{(see [NJ05])} \\ &\leq \frac{1}{\sigma^4} \sum_i (1-p_i)p_i = \frac{1}{\sigma^2}. \end{aligned}$$

  Hence, $\mathrm{Var}[\hat{\sigma}^2] = \sigma^4 \left(\frac{2}{m-1} + \frac{\kappa}{m} \right) \leq \frac{\sigma^4}{m}(4 + \frac{1}{\sigma^2})$. So Chebyshev's inequality implies that

  $$\Pr\left[|\hat{\sigma}^2 - \sigma^2| \geq t \frac{\sigma^2}{\sqrt{m}} \sqrt{4 + \frac{1}{\sigma^2}} \right] \leq \frac{1}{t^2}.$$

  Choosing $t = \sqrt{3}$ and $m = \lceil 3/\epsilon^2 \rceil$, the above imply that $|\hat{\sigma}^2 - \sigma^2| \leq \epsilon \sigma^2 \sqrt{4 + \frac{1}{\sigma^2}}$ with probability at least $2/3$.

□

We proceed to prove Lemma 5. `Learn-Poisson`$^X(n, \epsilon', \delta')$ runs $\mathcal{A}(n, \epsilon, \delta)$ from Lemma 6 with appropriately chosen $\epsilon = \epsilon(\epsilon')$ and $\delta = \delta(\delta')$, given below, and then outputs the translated Poisson distribution $TP(\hat{\mu}, \hat{\sigma}^2)$, where $\hat{\mu}$ and $\hat{\sigma}^2$ are the estimated mean and variance of X output by $\mathcal{A}$. Next, we show how to choose ϵ and δ, as well as why the desired guarantees are satisfied by the output distribution.

If X is not ϵ'-close to any PBD in sparse form inside the cover $\mathcal{S}_{\epsilon'}$ of Theorem 4, there exists a PBD Z in $(k = O(1/\epsilon'))$-heavy Binomial form inside $\mathcal{S}_{\epsilon'}$ that is within total variation distance ϵ' from X. We use the existence of such Z to obtain lower bounds on the mean and variance of X. Indeed, suppose that the distribution of Z is $\mathrm{Bin}(\ell, q)$, a Binomial with parameters ℓ, q. Then Theorem 4 certifies that the following conditions are satisfied by the parameters ℓ, q, $\mu = \mathbf{E}[X]$ and $\sigma^2 = \mathrm{Var}[X]$:

(a) $\ell q \geq k^2$;

(b) $\ell q(1-q) \geq k^2 - k - 1$;

(c) $|\ell q - \mu| = O(1)$; and

(d) $|\ell q(1-q) - \sigma^2| = O(1 + \epsilon' \cdot (1 + \sigma^2))$.

In particular, conditions (b) and (d) above imply that

$$\sigma^2 = \Omega(k^2) = \Omega(1/\epsilon'^2) \geq \theta^2,$$

for some universal constant θ, establishing (1). In terms of this θ, we choose $\epsilon = \epsilon'/\sqrt{4 + \frac{1}{\theta^2}}$ and $\delta = \delta'$ for the application of Lemma 6 to obtain—from $O(\log(1/\delta')/\epsilon'^2)$ samples—estimates $\hat{\mu}$ and $\hat{\sigma}^2$ of μ and σ^2.

From our choice of parameters and the guarantees of Lemma 6, it follows that, if X is not ϵ'-close to any PBD in sparse form inside the cover $\mathcal{S}_{\epsilon'}$, then with probability at least $1 - \delta'$ the estimates $\hat{\mu}$ and $\hat{\sigma}^2$ satisfy:

$$|\mu - \hat{\mu}| \leq \epsilon' \cdot \sigma \quad \text{and} \quad |\sigma^2 - \hat{\sigma}^2| \leq \epsilon' \cdot \sigma^2,$$

establishing (2). Moreover, if Y is a random variable distributed according to the translated Poisson distribution $TP(\hat{\mu}, \hat{\sigma}^2)$, we show that X and Y are within $O(\epsilon')$ in total variation distance, concluding the proof of Lemma 5.

Claim 7. *If X and Y are as above, then $d_{\mathrm{TV}}(X, Y) \leq O(\epsilon')$.*

Proof. We make use of Lemma 1. Suppose that $X = \sum_{i=1}^{n} X_i$, where $\mathbf{E}[X_i] = p_i$ for all i. Lemma 1 implies that

$$
\begin{aligned}
d_{\mathrm{TV}}(X, TP(\mu, \sigma^2)) &\leq \frac{\sqrt{\sum_i p_i^3(1-p_i)} + 2}{\sum_i p_i(1-p_i)} \\
&\leq \frac{\sqrt{\sum_i p_i(1-p_i)} + 2}{\sum_i p_i(1-p_i)} \\
&\leq \frac{1}{\sqrt{\sum_i p_i(1-p_i)}} + \frac{2}{\sum_i p_i(1-p_i)} \\
&= \frac{1}{\sigma} + \frac{2}{\sigma^2} \\
&= O(\epsilon'). \qquad (3)
\end{aligned}
$$

It remains to bound the total variation distance between the translated Poisson distributions $TP(\mu, \sigma^2)$ and $TP(\hat{\mu}, \hat{\sigma}^2)$. For this we use Lemma 2. Lemma 2 implies

$$
\begin{aligned}
d_{\mathrm{TV}}(TP(\mu, \sigma^2), TP(\hat{\mu}, \hat{\sigma}^2)) &\leq \frac{|\mu - \hat{\mu}|}{\min(\sigma, \hat{\sigma})} + \frac{|\sigma^2 - \hat{\sigma}^2| + 1}{\min(\sigma^2, \hat{\sigma}^2)} \\
&\leq \frac{\epsilon' \sigma}{\min(\sigma, \hat{\sigma})} + \frac{\epsilon' \cdot \sigma^2 + 1}{\min(\sigma^2, \hat{\sigma}^2)} \\
&\leq \frac{\epsilon' \sigma}{\sigma / \sqrt{1 - \epsilon'}} + \frac{\epsilon' \cdot \sigma^2 + 1}{\sigma^2/(1 - \epsilon')} \\
&= O(\epsilon') + \frac{O(1 - \epsilon')}{\sigma^2} \\
&= O(\epsilon') + O(\epsilon'^2) \\
&= O(\epsilon'). \qquad (4)
\end{aligned}
$$

The claim follows from (3), (4) and the triangle inequality. □

The proof of Lemma 5 is concluded. We remark that the algorithm described above does not need to know a priori whether or not X is ϵ'-close to a PBD in sparse form inside the cover $\mathcal{S}_{\epsilon'}$ of Theorem 4. The algorithm simply runs the estimator of Lemma 6 with $\epsilon = \epsilon' / \sqrt{4 + \frac{1}{\theta^2}}$ and $\delta' = \delta$ and outputs whatever estimates $\hat{\mu}$ and $\hat{\sigma}^2$ the algorithm of Lemma 6 produces. □

2.3 Hypothesis testing.

Our hypothesis testing routine $\texttt{Choose-Hypothesis}^X$ uses samples from the unknown distribution X to run a "competition" between two candidate hypothesis distributions H_1 and H_2 over $[n]$ that are given in the input. We show that if at least one of the two candidate hypotheses is close to the unknown distribution X, then with high probability over the samples drawn from X the routine selects as winner a candidate that is close to X. This basic approach of running a competition between candidate hypotheses is quite similar to the "Scheffé estimate" proposed by Devroye and Lugosi (see [DL96b, DL96a] and Chapter 6 of [DL01], as well as [Yat85]), but our notion of competition here is different.

We obtain the following lemma, postponing all running-time analysis to the next section.

Lemma 8. *There is an algorithm* Choose-Hypothesis$^X(H_1, H_2, \epsilon', \delta')$ *which is given sample access to distribution X, two hypothesis distributions H_1, H_2 for X, an accuracy parameter $\epsilon' > 0$, and a confidence parameter $\delta' > 0$. It makes*

$$m = O(\log(1/\delta')/\epsilon'^2)$$

draws from X and returns some $H \in \{H_1, H_2\}$. If $d_{\mathrm{TV}}(H_i, X) \leq \epsilon'$ for some $i \in \{1, 2\}$, then with probability at least $1 - \delta'$ the distribution H that Choose-Hypothesis *returns has $d_{\mathrm{TV}}(H, X) \leq 6\epsilon'$.*

Proof of Lemma 8: We first describe how the competition between H_1 and H_2 is carried out.

```
Choose-Hypothesis
INPUT: Sample access to distribution $X$; a pair of hypothesis distributions $(H_1, H_2)$; $\epsilon', \delta' > 0$.

Let $\mathcal{W}$ be the support of $X$, $\mathcal{W}_1 = \mathcal{W}_1(H_1, H_2) := \{w \in \mathcal{W} \,|\, H_1(w) > H_2(w)\}$, and $p_1 = H_1(\mathcal{W}_1)$, $p_2 = H_2(\mathcal{W}_1)$. /* Clearly, $p_1 > p_2$ and $d_{\mathrm{TV}}(H_1, H_2) = p_1 - p_2$. */

  1. If $p_1 - p_2 \leq 5\epsilon'$, declare a draw and return either $H_i$. Otherwise:
  2. Draw $m = 2\frac{\log(1/\delta')}{\epsilon'^2}$ samples $s_1, \ldots, s_m$ from $X$, and let $\tau = \frac{1}{m}|\{i \mid s_i \in \mathcal{W}_1\}|$ be the fraction of samples that fall inside $\mathcal{W}_1$.
  3. If $\tau > p_1 - \frac{3}{2}\epsilon'$, declare $H_1$ as winner and return $H_1$; otherwise,
  4. if $\tau < p_2 + \frac{3}{2}\epsilon'$, declare $H_2$ as winner and return $H_2$; otherwise,
  5. declare a draw and return either $H_i$.
```

The correctness of Choose-Hypothesis is an immediate consequence of the following claim. (In fact for Lemma 8 we only need item (i) below, but item (ii) will be handy later in the proof of Lemma 10.)

Claim 9. *Suppose that $d_{\mathrm{TV}}(X, H_i) \leq \epsilon'$, for some $i \in \{1, 2\}$. Then:*

(i) if $d_{\mathrm{TV}}(X, H_{3-i}) > 6\epsilon'$, the probability that Choose-Hypothesis$^X(H_1, H_2, \epsilon', \delta')$ *does not declare H_i as the winner is at most $2e^{-m\epsilon'^2/2}$, where m is chosen as in the description of the algorithm. (Intuitively, if H_{3-i} is very bad then it is very likely that H_i will be declared winner.)*

(ii) if $d_{\mathrm{TV}}(X, H_{3-i}) > 4\epsilon'$, the probability that Choose-Hypothesis$^X(H_1, H_2, \epsilon', \delta')$ *declares H_{3-i} as the winner is at most $2e^{-m\epsilon'^2/2}$. (Intuitively, if H_{3-i} is only moderately bad then a draw is possible but it is very unlikely that H_{3-i} will be declared winner.)*

Proof. Let $r = X(\mathcal{W}_1)$. The definition of the total variation distance implies that $|r - p_i| \leq \epsilon'$. Let us define independent indicators $\{Z_j\}_{j=1}^m$ such that, for all j, $Z_j = 1$ iff $s_j \in \mathcal{W}_1$. Clearly, $\tau = \frac{1}{m}\sum_{j=1}^m Z_j$ and $\mathbb{E}[\tau] = \mathbb{E}[Z_j] = r$. Since the Z_j's are mutually independent, it follows from the Chernoff bound that $\Pr[|\tau - r| \geq \epsilon'/2] \leq 2e^{-m\epsilon'^2/2}$. Using $|r - p_i| \leq \epsilon'$ we get that $\Pr[|\tau - p_i| \geq 3\epsilon'/2] \leq 2e^{-m\epsilon'^2/2}$. Hence:

- For part (i): If $d_{\mathrm{TV}}(X, H_{3-i}) > 6\epsilon'$, from the triangle inequality we get that $p_1 - p_2 = d_{\mathrm{TV}}(H_1, H_2) > 5\epsilon'$. Hence, the algorithm will go beyond step 1, and with probability at least $1 - 2e^{-m\epsilon'^2/2}$, it will stop at step 3 (when $i = 1$) or step 4 (when $i = 2$), declaring H_i as the winner of the competition between H_1 and H_2.

- For part (ii): If $p_1 - p_2 \leq 5\epsilon'$ then the competition declares a draw, hence H_{3-i} is not the winner. Otherwise we have $p_1 - p_2 > 5\epsilon'$ and the above arguments imply that the competition between H_1 and H_2 will declare H_{3-i} as the winner with probability at most $2e^{-m\epsilon'^2/2}$.

This concludes the proof of Claim 9. □

In view of Claim 9, the proof of Lemma 8 is concluded. □

Our Choose-Hypothesis algorithm implies a generic learning algorithm of independent interest.

Lemma 10. *Let $\mathcal{S}$ be an arbitrary set of distributions over a finite domain. Moreover, let $\mathcal{S}_\epsilon \subseteq \mathcal{S}$ be an ϵ-cover of $\mathcal{S}$ of size N, for some $\epsilon > 0$. For all $\delta > 0$, there is an algorithm that uses*

$$O(\epsilon^{-2} \log N \log(1/\delta))$$

samples from an unknown distribution $X \in \mathcal{S}$ and, with probability at least $1 - \delta$, outputs a distribution $Z \in \mathcal{S}_\epsilon$ that satisfies $d_{\mathrm{TV}}(X, Z) \leq 6\epsilon$.

Proof. The algorithm performs a tournament, by running $\texttt{Choose-Hypothesis}^X(H_i, H_j, \epsilon, \delta/(4N))$ for every pair (H_i, H_j), $i < j$, of distributions in $\mathcal{S}_\epsilon$. Then it outputs any distribution $Y^\star \in \mathcal{S}_\epsilon$ that was never a loser (i.e., won or tied against all other distributions in the cover). If no such distribution exists in $\mathcal{S}_\epsilon$ then the algorithm says "failure," and outputs an arbitrary distribution from $\mathcal{S}_\epsilon$.

Since $\mathcal{S}_\epsilon$ is an ϵ-cover of $\mathcal{S}$, there exists some $Y \in \mathcal{S}_\epsilon$ such that $d_{\mathrm{TV}}(X, Y) \leq \epsilon$. We first argue that with high probability this distribution Y never loses a competition against any other $Y' \in \mathcal{S}_\epsilon$ (so the algorithm does not output "failure"). Consider any $Y' \in \mathcal{S}_\epsilon$. If $d_{\mathrm{TV}}(X, Y') > 4\epsilon$, by Claim 9(ii) the probability that Y loses to Y' is at most $2e^{-m\epsilon^2/2} \leq \frac{\delta}{2N}$. On the other hand, if $d_{\mathrm{TV}}(X, Y') \leq 4\epsilon$, the triangle inequality gives that $d_{\mathrm{TV}}(Y, Y') \leq 5\epsilon$ and thus Y draws against Y'. A union bound over all $N - 1$ distributions in $\mathcal{S}_\epsilon \setminus \{Y\}$ shows that with probability at least $1 - \delta/2$, the distribution Y never loses a competition.

We next argue that with probability at least $1 - \delta/2$, every distribution $Y' \in \mathcal{S}_\epsilon$ that never loses must be close to X. Fix a distribution Y' such that $d_{\mathrm{TV}}(Y', X) > 6\epsilon$. Lemma 9(i) implies that Y' loses to Y with probability at least $1 - 2e^{-m\epsilon^2/2} \geq 1 - \delta/(2N)$. A union bound gives that with probability at least $1 - \delta/2$, every distribution Y' that has $d_{\mathrm{TV}}(Y', X) > 6\epsilon$ loses some competition.

Thus, with overall probability at least $1 - \delta$, the tournament does not output "failure" and outputs some distribution $Y^\star$ such that $d_{\mathrm{TV}}(X, Y^\star) \leq 6\epsilon$. This proves the lemma. □

We note that Devroye and Lugosi (Chapter 7 of [DL01]) prove a similar result, but there are some differences. They also have all pairs of distributions in the cover compete against each other, but they use a different notion of competition between every pair. Moreover, their approach chooses a distribution in the cover that wins the maximum number of competitions, whereas our algorithm chooses a distribution that is never defeated (i.e., won or tied against all other distributions in the cover).

2.4 Proof of Theorem 1.

We first show Part (1) of the theorem, where the learning algorithm may output any distribution over $[n]$ and not necessarily a PBD. Our algorithm has the structure outlined in Figure 1 with the following modifications: (a) first, if the total variation distance to within which we want to learn X is ϵ, the second argument of both Learn-Sparse and Learn-Poisson is set to $\frac{\epsilon}{12\max\{c_1, c_2\}}$, where c_1 and c_2 are respectively the constants from Lemmas 3 and 5; (b) we replace the third step with $\texttt{Choose-Hypothesis}^X(H_S, \widehat{H_P}, \epsilon/8, \delta/3)$, where $\widehat{H_P}$ is defined in terms of H_P as described below; and (c) if Choose-Hypothesis returns H_S, then Learn-PBD also returns H_S, while if Choose-Hypothesis returns $\widehat{H_P}$, then Learn-PBD returns H_P.

Definition of $\widehat{H_P}$: $\widehat{H_P}$ is defined in terms of H_P and the support of H_S in three steps: (i) for all points i such that $H_S(i) = 0$, we let $\widehat{H_P}(i) = H_P(i)$; (ii) for all points i such that $H_S(i) \neq 0$, we describe in Appendix C an efficient deterministic algorithm that numerically approximates $H_P(i)$ to within an additive error of $\pm\epsilon/48s$, where $s = O(1/\epsilon^3)$ is the cardinality of the support of H_S. If $\widehat{H}_{P,i}$ is the approximation to $H_P(i)$ output by the algorithm, we set $\widehat{H_P}(i) = \max\{0, \widehat{H}_{P,i} - \epsilon/48s\}$; notice then that $H_P(i) - \epsilon/24s \leq \widehat{H_P}(i) \leq H_P(i)$; finally (iii) for an arbitrary point i such that $H_S(i) = 0$, we set $\widehat{H_P}(i) = 1 - \sum_{j \neq i} \widehat{H_P}(j)$, to make sure that $\widehat{H_P}$ is a probability distribution. Observe that $\widehat{H_P}$ satisfies $d_{\mathrm{TV}}(\widehat{H_P}, H_P) \le \epsilon/24$, and therefore $|d_{\mathrm{TV}}(\widehat{H_P}, X) - d_{\mathrm{TV}}(X, H_P)| \le \epsilon/24$. Hence, if $d_{\mathrm{TV}}(X, H_P) \le \frac{\epsilon}{12}$, then $d_{\mathrm{TV}}(X, \widehat{H_P}) \le \frac{\epsilon}{8}$ and, if $d_{\mathrm{TV}}(X, \widehat{H_P}) \le \frac{6\epsilon}{8}$, then $d_{\mathrm{TV}}(X, H_P) \le \epsilon$.

We remark that the reason why we do not wish to use H_P directly in `Choose-Hypothesis` is purely computational. In particular, since H_P is a translated Poisson distribution, we cannot compute its probabilities $H_P(i)$ exactly, and we need to approximate them. On the other hand, we need to make sure that using approximate values will not cause `Choose-Hypothesis` to make a mistake. Our $\widehat{H_P}$ is carefully defined so as to make sure that `Choose-Hypothesis` selects a probability distribution that is close to the unknown X, and that all probabilities that `Choose-Hypothesis` needs to compute can be computed without much overhead. In particular, we remark that, in running `Choose-Hypothesis`, we do not a priori compute the value of $\widehat{H_P}$ at every point; we do instead a lazy evaluation of $\widehat{H_P}$, as explained in the running-time analysis below.

We proceed now to the analysis of our modified algorithm `Learn-PBD`. The sample complexity bound and correctness of our algorithm are immediate consequences of Lemmas 3, 5 and 8, taking into account the precise choice of constants and the distance between H_P and $\widehat{H_P}$. Next, let us bound the running time. Lemmas 3 and 5 bound the running time of Steps 1 and 2 of the algorithm, so it remains to bound the running time of the `Choose-Hypothesis` step. Notice that $\mathcal{W}_1(H_S, \widehat{H_P})$ is a subset of the support of the distribution H_S. Hence to compute $\mathcal{W}_1(H_S, \widehat{H_P})$ it suffices to determine the probabilities $H_S(i)$ and $\widehat{H_P}(i)$ for every point i in the support of H_S. For every such i, $H_S(i)$ is explicitly given in the output of `Learn-Sparse`, so we only need to compute $\widehat{H_P}(i)$. It follows from Theorem 6 (Appendix C) that the time needed to compute $\widehat{H_P}(i)$ is $\tilde{O}(\log(1/\epsilon)^3 + \log(1/\epsilon) \cdot (\log n + \langle\hat{\mu}\rangle + \langle\hat{\sigma}^2\rangle))$. Since $\hat{\mu}$ and $\hat{\sigma}^2$ are output by `Learn-Poisson`, by inspection of that algorithm it is easy to see that they each have bit complexity at most $O(\log n + \log(1/\epsilon))$ bits. Hence, given that the support of H_S has cardinality $O(1/\epsilon^3)$, the overall time spent computing the probabilities $\widehat{H_P}(i)$ for every point i in the support of H_S is $\tilde{O}(\frac{1}{\epsilon^3}\log n)$. After $\mathcal{W}_1$ is computed, the computation of the values $p_1 = H_S(\mathcal{W}_1)$, $q_1 = \widehat{H_P}(\mathcal{W}_1)$ and $p_1 - q_1$ takes time linear in the data produced by the algorithm so far, as these computations merely involve adding and subtracting probabilities that have already been explicitly computed by the algorithm. Computing the fraction of samples from X that fall inside $\mathcal{W}_1$ takes time $O\left(\log n \cdot \log(1/\delta)/\epsilon^2\right)$ and the rest of `Choose-Hypothesis` takes time linear in the size of the data that have been written down so far. Hence the overall running time of our algorithm is $\tilde{O}(\frac{1}{\epsilon^3}\log n \log^2 \frac{1}{\delta})$. This gives Part (1) of Theorem 1.

Now we turn to Part (2) of Theorem 1, the proper learning result. We explain how to modify the algorithm of Part (1) to produce a PBD that is within $O(\epsilon)$ of the unknown X. The main modifications are the following. First, we replace `Learn-Sparse` with a different learning algorithm, `Proper-Learn-Sparse`, which is based on Lemma 10, and always outputs a PBD. Second, we add a post-processing step to `Learn-Poisson` that converts H_P to a PBD. After we describe these new ingredients in detail, we describe our proper learning algorithm.

1. `Proper-Learn-Sparse`$^X(n, \epsilon, \delta)$: This procedure draws $\tilde{O}(1/\epsilon^2) \cdot \log(1/\delta)$ samples from X, does $(1/\epsilon)^{O\left(\log^2(1/\epsilon)\right)} \cdot \tilde{O}\left(\log n \cdot \log \frac{1}{\delta}\right)$ bit operations, and outputs a PBD H_S in sparse form. The guarantee is similar to that of `Learn-Sparse`. Namely, if X is ϵ-close to some sparse form PBD Y in the cover $\mathcal{S}_\epsilon$ of Theorem 4, then, with probability at least $1-\delta$ over the samples drawn from X, $d_{\mathrm{TV}}(X, H_S) \le 6\epsilon$.

   We proceed to describe the procedure in tandem with a proof of correctness. As in `Learn-Sparse`, we start by truncating $\Theta(\epsilon)$ of the probability mass from each end of X to obtain a conditional distribution $X_{[\hat{a},\hat{b}]}$. In particular, we compute $\hat{a}$ and $\hat{b}$ as described in the beginning of the proof of Lemma 3 (setting $\epsilon' = \epsilon$ and $\delta' = \delta$). Claim 4 implies that, with probability at least $1 - \delta/2$, $X(\le \hat{a}), 1 - X(\le \hat{b}) \in [3\epsilon/2, 5\epsilon/2]$. (Let us denote this event by $\mathcal{G}$.) We distinguish the following cases:

   - If $\hat{b} - \hat{a} > \omega = (C/\epsilon)^3$, where C is the constant in the statement of Theorem 4, the algorithm outputs "fail," returning the trivial hypothesis that puts probability mass 1 on the point 0. Observe that, if

$\hat{b} - \hat{a} > \omega$ and $X(\leq \hat{a}), 1 - X(\leq \hat{b}) \in [3\epsilon/2, 5\epsilon/2]$, then X cannot be ϵ-close to a sparse-form distribution in the cover.

- If $\hat{b} - \hat{a} \leq \omega$, then the algorithm proceeds as follows. Let $\mathcal{S}'_\epsilon$ be an ϵ-cover of the set of all PBDs of order ω, i.e., all PBDs which are sums of just ω Bernoulli random variables. By Theorem 4, it follows that $|\mathcal{S}'_\epsilon| = (1/\epsilon)^{O(\log^2(1/\epsilon))}$ and that $\mathcal{S}'_\epsilon$ can be constructed in time $(1/\epsilon)^{O(\log^2(1/\epsilon))}$. Now, let $\tilde{\mathcal{S}}_\epsilon$ be the set of all distributions of the form $A(x - \beta)$ where A is a distribution from $\mathcal{S}'_\epsilon$ and β is an integer "shift" which is in the range $[\hat{a} - \omega, \ldots, \hat{b}]$. Observe that there are $O(1/\epsilon^3)$ possibilities for β and $|\mathcal{S}'_\epsilon|$ possibilities for A, so we similarly get that $|\tilde{\mathcal{S}}_\epsilon| = (1/\epsilon)^{O(\log^2(1/\epsilon)}$ and that $\tilde{\mathcal{S}}_\epsilon$ can be constructed in time $(1/\epsilon)^{O(\log^2(1/\epsilon)} \log n$. Our algorithm `Proper-Learn-Sparse` constructs the set $\tilde{\mathcal{S}}_\epsilon$ and runs the tournament described in the proof of Lemma 10 (using $\tilde{\mathcal{S}}_\epsilon$ in place of $\mathcal{S}_\epsilon$, and $\delta/2$ in place of δ). We will show that, if X is ϵ-close to some sparse form PBD $Y \in \mathcal{S}_\epsilon$ and event $\mathcal{G}$ happens, then, with probability at least $1 - \frac{\delta}{2}$, the output of the tournament is a sparse PBD that is 6ϵ-close to X.

Analysis: The sample complexity and running time of `Proper-Learn-Sparse` follow immediately from Claim 4 and Lemma 10. To show correctness, it suffices to argue that, if X is ϵ-close to some sparse form PBD $Y \in \mathcal{S}_\epsilon$ and event $\mathcal{G}$ happens, then X is ϵ-close to some distribution in $\tilde{\mathcal{S}}_\epsilon$. Indeed, suppose that Y is an order ω PBD Z translated by some β and suppose that $X(\leq \hat{a}), 1 - X(\leq \hat{b}) \in [3\epsilon/2, 5\epsilon/2]$. Since at least $1 - O(\epsilon)$ of the mass of X is in $[\hat{a}, \hat{b}]$, it is clear that β must be in the range $[\hat{a} - \omega, \ldots, \hat{b}]$, as otherwise X could not be ϵ-close to Y. So $Y \in \tilde{\mathcal{S}}_\epsilon$.

2. `Locate-Binomial`$(\hat{\mu}, \hat{\sigma}^2, n)$: This routine takes as input the output $(\hat{\mu}, \hat{\sigma}^2)$ of `Learn-Poisson`$^X(n, \epsilon, \delta)$ and computes a Binomial distribution H_B, without any additional samples from X. The guarantee is that, if X is not ϵ-close to any sparse form distribution in the cover S_ϵ of Theorem 4, then, with probability at least $1 - \delta$ (over the randomness in the output of `Learn-Poisson`), H_B will be $O(\epsilon)$-close to X.

   Let μ and σ^2 be the (unknown) mean and variance of distribution X and assume that X is not ϵ-close to any sparse form distribution in S_ϵ. Our analysis from Section 2.2 shows that, with probability at least $1 - \delta$, the output $(\hat{\mu}, \hat{\sigma}^2)$ of `Learn-Poisson`$^X(n, \epsilon, \delta)$ satisfies that $d_{\mathrm{TV}}(X, TP(\hat{\mu}, \hat{\sigma}^2)) = O(\epsilon)$ as well as the bounds (1) and (2) of Section 2.2 (with ϵ in place of ϵ'). We will call all these conditions our "working assumptions." We provide no guarantees when the working assumptions are not satisfied.

   We proceed to describe `Locate-Binomial`. Our routine has three steps. The first two eliminate corner-cases in the values of $\hat{\mu}$ and $\hat{\sigma}^2$, while the last step defines a Binomial distribution $H_B \equiv \mathrm{Bin}(\hat{n}, \hat{p})$ with $\hat{n} \leq n$ that is $O(\epsilon)$-close to $H_P \equiv TP(\hat{\mu}, \hat{\sigma}^2)$ and hence to X under our working assumptions. (We note that a significant portion of the work below is to ensure that $\hat{n} \leq n$, which does not seem to follow from a more direct approach. Getting $\hat{n} \leq n$ is necessary in order for our learning algorithm for order-n PBDs to be truly proper.) Throughout (a), (b) and (c) below we assume that our working assumptions hold. In particular, our assumptions are used every time we employ the bounds (1) and (2) of Section 2.2.

   (a) Tweaking $\hat{\sigma}^2$: If $\hat{\sigma}^2 \leq \frac{n}{4}$, we set $\sigma_1^2 = \hat{\sigma}^2$; otherwise, we set $\sigma_1^2 = \frac{n}{4}$. We note for future reference that in both cases (2) gives
   $$(1 - \epsilon)\sigma^2 \leq \sigma_1^2 \leq (1 + \epsilon)\sigma^2, \tag{5}$$
   where the lower bound follows from (2) and the fact that any PBD satisfies $\sigma^2 \leq \frac{n}{4}$.
   We prove next that our setting of σ_1^2 results in $d_{\mathrm{TV}}(TP(\hat{\mu}, \hat{\sigma}^2), TP(\hat{\mu}, \sigma_1^2)) \leq O(\epsilon)$. Indeed, if $\hat{\sigma}^2 \leq \frac{n}{4}$ then this distance is zero and the claim certainly holds. Otherwise we have that $(1 + \epsilon)\,\sigma^2 \geq \hat{\sigma}^2 > \sigma_1^2 = \frac{n}{4} \geq \sigma^2$, where we used (2). Hence, by Lemma 2 we get:
   $$\begin{aligned} d_{\mathrm{TV}}(TP(\hat{\mu}, \hat{\sigma}^2), TP(\hat{\mu}, \sigma_1^2)) &\leq \frac{|\hat{\sigma}^2 - \sigma_1^2| + 1}{\hat{\sigma}^2} \\ &\leq \frac{\epsilon\sigma^2 + 1}{\sigma^2} = O(\epsilon), \end{aligned} \tag{6}$$

where we used the fact that $\sigma^2 = \Omega(1/\epsilon^2)$ from (1).

(b) Tweaking σ_1^2: If $\hat{\mu}^2 \le n(\hat{\mu} - \sigma_1^2)$, set $\sigma_2^2 = \sigma_1^2$; otherwise, set $\sigma_2^2 = \frac{n\hat{\mu} - \hat{\mu}^2}{n}$. We claim that this results in $d_{\mathrm{TV}}(TP(\hat{\mu}, \sigma_1^2), TP(\hat{\mu}, \sigma_2^2)) \le O(\epsilon)$. Indeed, if $\hat{\mu}^2 \le n(\hat{\mu} - \sigma_1^2)$, then clearly the distance is zero and the claim holds. Otherwise

- Observe first that $\sigma_1^2 > \sigma_2^2$ and $\sigma_2^2 \ge 0$, where the last assertion follows from the fact that $\hat{\mu} \le n$ by construction.
- Next, suppose that $X = PBD(p_1, \ldots, p_n)$. Then from Cauchy-Schwarz we get that

$$\mu^2 = \left(\sum_{i=1}^{n} p_i\right)^2 \le n\left(\sum_{i=1}^{n} p_i^2\right) = n(\mu - \sigma^2).$$

Rearranging this yields

$$\frac{\mu(n-\mu)}{n} \ge \sigma^2. \tag{7}$$

We now have that

$$\begin{aligned}\sigma_2^2 = \frac{n\hat{\mu} - \hat{\mu}^2}{n} &\ge \frac{n(\mu - \epsilon\sigma) - (\mu + \epsilon\sigma)^2}{n} \\ &= \frac{n\mu - \mu^2 - \epsilon^2\sigma^2 - \epsilon\sigma(n + 2\mu)}{n} \\ &\ge \sigma^2 - \frac{\epsilon^2}{n}\sigma^2 - 3\epsilon\sigma \\ &\ge (1 - \epsilon^2)\sigma^2 - 3\epsilon\sigma \ge (1 - O(\epsilon))\sigma^2 \end{aligned} \tag{8}$$

where the first inequality follows from (2), the second inequality follows from (7) and the fact that any PBD over n variables satisfies $\mu \le n$, and the last one from (1).

- Given the above, we get by Lemma 2 that:

$$\begin{aligned} d_{\mathrm{TV}}(TP(\hat{\mu}, \sigma_1^2), TP(\hat{\mu}, \sigma_2^2)) &\le \frac{\sigma_1^2 - \sigma_2^2 + 1}{\sigma_1^2} \\ &\le \frac{(1+\epsilon)\sigma^2 - (1 - O(\epsilon))\sigma^2 + 1}{(1-\epsilon)\sigma^2} = O(\epsilon), \end{aligned} \tag{9}$$

where we used that $\sigma^2 = \Omega(1/\epsilon^2)$ from (1).

(c) Constructing a Binomial Distribution: We construct a Binomial distribution H_B that is $O(\epsilon)$-close to $TP(\hat{\mu}, \sigma_2^2)$. If we do this then, by (6), (9), our working assumption that $d_{\mathrm{TV}}(H_P, X) = O(\epsilon)$, and the triangle inequality, we have that $d_{\mathrm{TV}}(H_B, X) = O(\epsilon)$ and we are done. The Binomial distribution H_B that we construct is $\mathrm{Bin}(\hat{n}, \hat{p})$, where

$$\hat{n} = \left\lfloor \hat{\mu}^2 / (\hat{\mu} - \sigma_2^2) \right\rfloor \text{ and } \hat{p} = (\hat{\mu} - \sigma_2^2)/\hat{\mu}.$$

Note that, from the way that σ_2^2 is set in Step (b) above, we have that $\hat{n} \le n$ and $\hat{p} \in [0, 1]$, as required for $\mathrm{Bin}(\hat{n}, \hat{p})$ to be a valid Binomial distribution and a valid output for Part 2 of Theorem 1.

Let us bound the total variation distance between $\mathrm{Bin}(\hat{n}, \hat{p})$ and $TP(\hat{\mu}, \sigma_2^2)$. First, using Lemma 1 we have:

$$\begin{aligned} & d_{\mathrm{TV}}(\mathrm{Bin}(\hat{n}, \hat{p}), TP(\hat{n}\hat{p}, \hat{n}\hat{p}(1-\hat{p})) \\ \le\ & \frac{1}{\sqrt{\hat{n}\hat{p}(1-\hat{p})}} + \frac{2}{\hat{n}\hat{p}(1-\hat{p})}. \end{aligned} \tag{10}$$

Notice that

$$\begin{aligned}\hat{n}\hat{p}(1-\hat{p}) &\geq \left(\frac{\hat{\mu}^2}{\hat{\mu}-\sigma_2^2}-1\right)\left(\frac{\hat{\mu}-\sigma_2^2}{\hat{\mu}}\right)\left(\frac{\sigma_2^2}{\hat{\mu}}\right)\\ &= \sigma_2^2-\hat{p}(1-\hat{p}) \geq (1-O(\epsilon))\sigma^2-1\\ &\geq \Omega(1/\epsilon^2),\end{aligned}$$

where the second inequality uses (8) (or (5) depending on which case of Step (b) we fell into) and the last one uses the fact that $\sigma^2 = \Omega(1/\epsilon^2)$ from (1). So plugging this into (10) we get:

$$d_{\mathrm{TV}}(\mathrm{Bin}(\hat{n},\hat{p}), TP(\hat{n}\hat{p}, \hat{n}\hat{p}(1-\hat{p})) = O(\epsilon).$$

The next step is to compare $TP(\hat{n}\hat{p}, \hat{n}\hat{p}(1-\hat{p}))$ and $TP(\hat{\mu}, \sigma_2^2)$. Lemma 2 gives:

$$\begin{aligned}&d_{\mathrm{TV}}(TP(\hat{n}\hat{p}, \hat{n}\hat{p}(1-\hat{p})), TP(\hat{\mu}, \sigma_2^2))\\ \leq\ & \frac{|\hat{n}\hat{p}-\hat{\mu}|}{\min(\sqrt{\hat{n}\hat{p}(1-\hat{p})}, \sigma_2)} + \frac{|\hat{n}\hat{p}(1-\hat{p})-\sigma_2^2|+1}{\min(\hat{n}\hat{p}(1-\hat{p}), \sigma_2^2)}\\ \leq\ & \frac{1}{\sqrt{\hat{n}\hat{p}(1-\hat{p})}} + \frac{2}{\hat{n}\hat{p}(1-\hat{p})}\\ =\ & O(\epsilon).\end{aligned}$$

By the triangle inequality we get

$$d_{\mathrm{TV}}(\mathrm{Bin}(\hat{n},\hat{p}), TP(\hat{\mu}, \sigma_2^2) = O(\epsilon),$$

which was our ultimate goal.

3. `Proper-Learn-PBD`: Given the `Proper-Learn-Sparse` and `Locate-Binomial` routines described above, we are ready to describe our proper learning algorithm. The algorithm is similar to our non-proper learning one, `Learn-PBD`, with the following modifications: In the first step, instead of running `Learn-Sparse`, we run `Proper-Learn-Sparse` to get a sparse from PBD H_S. In the second step, we still run `Learn-Poisson` as we did before to get a translated Poisson distribution H_P. Then we run `Choose-Hypothesis` feeding it H_S and H_P as input. If the distribution returned by `Choose-Hypothesis` is H_S, we just output H_S. If it returns H_P instead, then we run `Locate-Binomial` to convert it to a Binomial distribution that is still close to the unknown distribution X. We tune the parameters ϵ and δ based on the above analyses to guarantee that, with probability at least $1-\delta$, the distribution output by our overall algorithm is ϵ-close to the unknown distribution X. The number of samples we need is $\tilde{O}(1/\epsilon^2)\log(1/\delta)$, and the running time is $\left(\frac{1}{\epsilon}\right)^{O(\log^2 1/\epsilon)} \cdot \tilde{O}(\log n \cdot \log \frac{1}{\delta})$. This concludes the proof of Part 2 of Theorem 1, and thus of the entire theorem. □

3 Learning weighted sums of independent Bernoullis

In this section we consider a generalization of the problem of learning an unknown PBD, by studying the learnability of weighted sums of independent Bernoulli random variables $X = \sum_{i=1}^n w_i X_i$. (Throughout this section we assume for simplicity that the weights are "known" to the learning algorithm.) In Section 3.1 we show that if there are only constantly many different weights then such distributions can be learned by an algorithm that uses $O(\log n)$ samples and runs in time $\mathrm{poly}(n)$. In Section 3.2 we show that if there are n distinct weights then even if those weights have an extremely simple structure – the i-th weight is simply i – any algorithm must use $\Omega(n)$ samples.

3.1 Learning sums of weighted independent Bernoulli random variables with few distinct weights

Recall Theorem 2:

THEOREM 2. *Let $X = \sum_{i=1}^{n} a_i X_i$ be a weighted sum of unknown independent Bernoulli random variables such that there are at most k different values in the set $\{a_1, \ldots, a_n\}$. Then there is an algorithm with the following properties: given $n, a_1, \ldots, a_n$ and access to independent draws from X, it uses*

$$\widetilde{O}(k/\epsilon^2) \cdot \log(n) \cdot \log(1/\delta)$$

samples from the target distribution X, runs in time

$$\text{poly}\left(n^k \cdot (k/\epsilon)^{k\log^2(k/\epsilon)}\right) \cdot \log(1/\delta),$$

and with probability at least $1-\delta$ outputs a hypothesis vector $\hat{p} \in [0,1]^n$ defining independent Bernoulli random variables $\hat{X}_i$ with $\mathbf{E}[\hat{X}_i] = p_i$ such that $d_{\mathrm{TV}}(\hat{X}, X) \leq \epsilon$, where $\hat{X} = \sum_{i=1}^{n} a_i \hat{X}_i$.

Given a vector $\overline{a} = (a_1, \ldots, a_n)$ of weights, we refer to a distribution $X = \sum_{i=1}^{n} a_i X_i$ (where $X_1, \ldots, X_n$ are independent Bernoullis which may have arbitrary means) as an *$\overline{a}$-weighted sum of Bernoullis*, and we write $\mathcal{S}_{\overline{a}}$ to denote the space of all such distributions.

To prove Theorem 2 we first show that $\mathcal{S}_{\overline{a}}$ has an ϵ-cover that is not too large. We then show that by running a "tournament" between all pairs of distributions in the cover, using the hypothesis testing subroutine from Section 2.3, it is possible to identify a distribution in the cover that is close to the target $\overline{a}$-weighted sum of Bernoullis.

Lemma 11. *There is an ϵ-cover $\mathcal{S}_{\overline{a},\epsilon} \subset \mathcal{S}_{\overline{a}}$ of size $|\mathcal{S}_{\overline{a},\epsilon}| \leq (n/k)^{3k} \cdot (k/\epsilon)^{k \cdot O(\log^2(k/\epsilon))}$ that can be constructed in time $\text{poly}(|\mathcal{S}_{\overline{a},\epsilon}|)$.*

Proof. Let $\{b_j\}_{j=1}^{k}$ denote the set of distinct weights in $a_1, \ldots, a_n$, and let $n_j = \left|\{i \in [n] \mid a_i = b_j\}\right|$. With this notation, we can write $X = \sum_{j=1}^{k} b_j S_j = g(S)$, where $S = (S_1, \ldots, S_k)$ with each S_j a sum of n_j many independent Bernoulli random variables and $g(y_1, \ldots, y_k) = \sum_{j=1}^{k} b_j y_j$. Clearly we have $\sum_{j=1}^{k} n_j = n$. By Theorem 4, for each $j \in \{1, \ldots, k\}$ the space of all possible S_j's has an explicit (ϵ/k)-cover $\mathcal{S}_{\epsilon/k}^{j}$ of size $|\mathcal{S}_{\epsilon/k}^{j}| \leq n_j^2 + n \cdot (k/\epsilon)^{O(\log^2(k/\epsilon))}$. By independence across S_j's, the product $\mathcal{Q} = \prod_{j=1}^{k} \mathcal{S}_{\epsilon/k}^{j}$ is an ϵ-cover for the space of all possible S's, and hence the set

$$\{Q = \sum_{j=1}^{k} b_j S_j \ : \ (S_1, \ldots, S_k) \in \mathcal{Q}\}$$

is an ϵ-cover for $\mathcal{S}_{\overline{a}}$. So $\mathcal{S}_{\overline{a}}$ has an explicit ϵ-cover of size $|\mathcal{Q}| = \prod_{j=1}^{k} |\mathcal{S}_{\epsilon/k}^{j}| \leq (n/k)^{2k} \cdot (k/\epsilon)^{k \cdot O(\log^2(k/\epsilon))}$. □

Proof of Theorem 2: We claim that the algorithm of Lemma 10 has the desired sample complexity and can be implemented to run in the claimed time bound. The sample complexity bound follows directly from Lemma 10. It remains to argue about the time complexity. Note that the running time of the algorithm is $\text{poly}(|\mathcal{S}_{\overline{a},\epsilon}|)$ times the running time of a competition. We will show that a competition between $H_1, H_2 \in \mathcal{S}_{\overline{a},\epsilon}$ can be carried out by an efficient algorithm. This amounts to efficiently computing the probabilities $p_1 = H_1(\mathcal{W}_1)$ and $q_1 = H_2(\mathcal{W}_1)$ and efficiently computing $H_1(x)$ and $H_2(x)$ for each of the m samples x drawn in step (2) of the competition. Note that $\mathcal{W} = \sum_{j=1}^{k} b_i \cdot \{0, 1, \ldots, n_j\}$. Clearly, $|\mathcal{W}| \leq \prod_{j=1}^{k}(n_j + 1) = O((n/k)^k)$. It is thus easy to see that p_1, q_1 and each of $H_1(x), H_2(x)$ can be efficiently computed as long as there is an efficient algorithm for the following problem: given $H = \sum_{j=1}^{k} b_j S_j \in \mathcal{S}_{\overline{a},\epsilon}$ and $w \in \mathcal{W}$, compute $H(w)$. Indeed, fix any such H, w. We have that

$$H(w) = \sum_{m_1, \ldots, m_k} \prod_{j=1}^{k} \Pr_{H}[S_j = m_j],$$

where the sum is over all k-tuples $(m_1, \ldots, m_k)$ such that $0 \le m_j \le n_j$ for all j and $b_1 m_1 + \cdots + b_k m_k = w$ (as noted above there are at most $O((n/k)^k)$ such k-tuples). To complete the proof of Theorem 2 we note that $\Pr_H[S_j = m_j]$ can be computed in $O(n_j^2)$ time by standard dynamic programming. □

We close this subsection with the following remark: In [DDS12b] the authors have given a $\mathrm{poly}(\ell, \log(n), 1/\epsilon)$-time algorithm that learns any ℓ-modal distribution over $[n]$ (i.e., a distribution whose pdf has at most ℓ "peaks" and "valleys") using $O(\ell \log(n)/\epsilon^3 + (\ell/\epsilon)^3 \log(\ell/\epsilon))$ samples. It is natural to wonder whether this algorithm could be used to efficiently learn a sum of n weighted independent Bernoulli random variables with k distinct weights, and thus give an alternate algorithm for Theorem 2, perhaps with better asymptotic guarantees. However, it is easy to construct a sum $X = \sum_{i=1}^n a_i X_i$ of n weighted independent Bernoulli random variables with k distinct weights such that X is 2^k-modal. Thus, a naive application of the [DDS12b] result would only give an algorithm with sample complexity exponential in k, rather than the quasilinear sample complexity of our current algorithm. If the 2^k-modality of the above-mentioned example is the worst case (which we do not know), then the [DDS12b] algorithm would give a $\mathrm{poly}(2^k, \log(n), 1/\epsilon)$-time algorithm for our problem that uses $O(2^k \log(n)/\epsilon^3) + 2^{O(k)} \cdot \tilde{O}(1/\epsilon^3)$ examples (so comparing with Theorem 2, exponentially worse sample complexity as a function of k, but exponentially better running time as a function of n). Finally, in the context of this question (how many modes can there be for a sum of n weighted independent Bernoulli random variables with k distinct weights), it is interesting to recall the result of K.-I. Sato [Sat93] which shows that for any N there are two unimodal distributions X, Y such that $X + Y$ has at least N modes.

3.2 Sample complexity lower bound for learning sums of weighted independent Bernoulli random variables

Recall Theorem 3:

THEOREM 3. *Let $X = \sum_{i=1}^n i \cdot X_i$ be a weighted sum of unknown independent Bernoulli random variables (where the i-th weight is simply i). Let L be any learning algorithm which, given n and access to independent draws from X, outputs a hypothesis distribution $\hat{X}$ such that $d_{\mathrm{TV}}(\hat{X}, X) \le 1/25$ with probability at least $e^{-o(n)}$. Then L must use $\Omega(n)$ samples.*

Proof of Theorem 3: We define a probability distribution over possible target probability distributions X as follows: A subset $S \subset \{n/2+1, \ldots, n\}$ of size $|S| = n/100$ is drawn uniformly at random from all $\binom{n/2}{n/100}$ possible outcomes.. The vector $\overline{p} = (p_1, \ldots, p_n)$ is defined as follows: for each $i \in S$ the value p_i equals $100/n = 1/|S|$, and for all other i the value p_i equals 0. The i-th Bernoulli random variable X_i has $\mathbf{E}[X_i] = p_i$, and the target distribution is $X = X_{\overline{p}} = \sum_{i=1}^n i X_i$.

We will need two easy lemmas:

Lemma 12. *Fix any $S, \overline{p}$ as described above. For any $j \in \{n/2+1, \ldots, n\}$ we have $X_{\overline{p}}(j) \neq 0$ if and only if $j \in S$. For any $j \in S$ the value $X_{\overline{p}}(j)$ is exactly $(100/n)(1-100/n)^{n/100-1} > 35/n$ (for n sufficiently large), and hence $X_{\overline{p}}(\{n/2+1, \ldots, n\}) > 0.35$ (again for n sufficiently large).*

The first claim of the lemma holds because any set of $c \ge 2$ numbers from $\{n/2+1, \ldots, n\}$ must sum to more than n. The second claim holds because the only way a draw x from $X_{\overline{p}}$ can have $x = j$ is if $X_j = 1$ and all other X_i are 0 (here we are using $\lim_{x \to \infty}(1 - 1/x)^x = 1/e$).

The next lemma is an easy consequence of Chernoff bounds:

Lemma 13. *Fix any $\overline{p}$ as defined above, and consider a sequence of $n/2000$ independent draws from $X_{\overline{p}} = \sum_i i X_i$. With probability $1 - e^{-\Omega(n)}$ the total number of indices $j \in [n]$ such that X_j is ever 1 in any of the $n/2000$ draws is at most $n/1000$.*

We are now ready to prove Theorem 3. Let L be a learning algorithm that receives $n/2000$ samples. Let $S \subset \{n/2+1, \ldots, n\}$ and $\overline{p}$ be chosen randomly as defined above, and set the target to $X = X_{\overline{p}}$.

We consider an augmented learner L' that is given "extra information." For each point in the sample, instead of receiving the value of that draw from X the learner L' is given the entire vector $(X_1, \ldots, X_n) \in \{0,1\}^n$. Let T denote the set of elements $j \in \{n/2+1, \ldots, n\}$ for which the learner is ever given a vector $(X_1, \ldots, X_n)$ that has $X_j = 1$. By Lemma 13 we have $|T| \leq n/1000$ with probability at least $1 - e^{-\Omega(n)}$; we condition on the event $|T| \leq n/1000$ going forth.

Fix any value $\ell \leq n/1000$. Conditioned on $|T| = \ell$, the set T is equally likely to be any ℓ-element subset of S, and all possible "completions" of T with an additional $n/100 - \ell \geq 9n/1000$ elements of $\{n/2+1, \ldots, n\} \setminus T$ are equally likely to be the true set S.

Let H denote the hypothesis distribution over $[n]$ that algorithm L outputs. Let R denote the set $\{n/2+1, \ldots, n\} \setminus T$; note that since $|T| = \ell \leq n/1000$, we have $|R| \geq 499n/1000$. Let U denote the set $\{i \in R \ : \ H(i) \geq 30/n\}$. Since H is a distribution we must have $|U| \leq n/30$. It is easy to verify that we have $d_{\mathrm{TV}}(X, H) \geq \frac{5}{n}|S \backslash U|$. Since S is a uniform random extension of T with at most $n/100 - \ell \in [9n/1000, n/100]$ unknown elements of R and $|R| \geq 499n/1000$, an easy calculation shows that $\Pr[|S \setminus U| > 8n/1000]$ is $1 - e^{-\Omega(n)}$. This means that with probability $1 - e^{-\Omega(n)}$ we have $d_{\mathrm{TV}}(X, H) \geq \frac{8n}{1000} \cdot \frac{5}{n} = 1/25$, and the theorem is proved. □

4 Conclusion and open problems

Since the initial conference publication of this work [DDS12a], some progress has been made on problems related to learning Poisson Binomial Distributions. The initial conference version [DDS12a] asked whether log-concave distributions over $[n]$ (a generalization of Poisson Binomial Distributions) can be learned to accuracy ϵ with $\mathrm{poly}(1/\epsilon)$ samples independent of n. An affirmative answer to this question was subsequently provided in [CDSS13]. More recently, [DDO+13] studied a different generalization of Poisson Binomial Distributions by considering random variables of the form $X = \sum_{i=1}^{n} X_i$ where the X_i's are mutually independent (not necessarily identical) distributions that are each supported on the integers $\{0, 1, \ldots, k-1\}$ (so, the $k = 2$ case corresponds to Poisson Binomial Distributions). [DDO+13] gave an algorithm for learning these distributions to accuracy ϵ using $\mathrm{poly}(k, 1/\epsilon)$ samples (independent of n).

While our results in this paper essentially settle the sample complexity of learning an unknown Poisson Binomial Distribution, several goals remain for future work. Our non-proper learning algorithm is computationally more efficient than our proper learning algorithm, but uses a factor of $1/\epsilon$ more samples. An obvious goal is to obtain "the best of both worlds" by coming up with an $O(1/\epsilon^2)$-sample algorithm which performs $\tilde{O}(\log(n)/\epsilon^2)$ bit operations and learns an unknown PBD to accuracy ϵ (ideally, such an algorithm would even be proper and output a PBD as its hypothesis). Another goal is to sharpen the sample complexity bounds of [DDO+13] and determine the correct polynomial dependence on k and $1/\epsilon$ for the generalized problem studied in that work.

References

[Ber41] Andrew C. Berry. The Accuracy of the Gaussian Approximation to the Sum of Independent Variates. *Transactions of the American Mathematical Society*, 49(1):122–136, 1941. 1.2

[BHJ92] A.D. Barbour, L. Holst, and S. Janson. *Poisson Approximation*. Oxford University Press, New York, NY, 1992. 1.2

[Bir87a] L. Birgé. Estimating a density under order restrictions: Nonasymptotic minimax risk. *Annals of Statistics*, 15(3):995–1012, 1987. 1.2, B

[Bir87b] L. Birgé. On the risk of histograms for estimating decreasing densities. *Annals of Statistics*, 15(3):1013–1022, 1987. B

[Bir97] L. Birgé. Estimation of unimodal densities without smoothness assumptions. *Annals of Statistics*, 25(3):970–981, 1997. 1.2, 2.1, 5, B

[BL06] A. D. Barbour and T. Lindvall. Translated Poisson Approximation for Markov Chains. *Journal of Theoretical Probability*, 19, 2006. 2

[Bre75] R. P. Brent. Multiple-precision zero-finding methods and the complexity of elementary function evaluation. *Analytic Computational Complexity (J. F. Traub ed.)*, pages 151–176, 1975. Academic Press, New York. C, 5

[Bre76] R. P. Brent. Fast multiple-precision evaluation of elementary functions. *Journal of the ACM*, 23(2):242–251, 1976. 3

[BS10] Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *FOCS*, pages 103–112, 2010. 1.2

[Cam60] L. Le Cam. An approximation theorem for the Poisson binomial distribution. *Pacific J. Math*, 10:1181–1197, 1960. 1.2

[CDSS13] S. Chan, I. Diakonikolas, R. Servedio, and X. Sun. Learning mixtures of structured distributions over discrete domains. In *SODA*, pages 1380–1394, 2013. 4

[Che52] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *Ann. Math. Statist.*, 23:493–507, 1952. 1

[Che74] L.H.Y. Chen. On the convergence of Poisson binomial to Poisson distributions. *Ann. Probab.*, 2:178–180, 1974. 1.2

[CL97] S.X. Chen and J.S. Liu. Statistical applications of the Poisson-Binomial and Conditional Bernoulli Distributions. *Statistica Sinica*, 7:875–892, 1997. 1

[Das08] Constantinos Daskalakis. An Efficient PTAS for Two-Strategy Anonymous Games. *WINE* 2008, pp. 186-197. Full version available as ArXiV report, 2008. 1.3

[DDO$^+$13] C. Daskalakis, I. Diakonikolas, R. O'Donnell, R. Servedio, and L.-Y. Tan. Learning Sums of Independent Integer Random Variables. In *FOCS*, 2013. 4

[DDS12a] C. Daskalakis, I. Diakonikolas, and R. Servedio. Learning Poisson Binomial Distributions. In *STOC*, pages 709–728, 2012. 4

[DDS12b] C. Daskalakis, I. Diakonikolas, and R.A. Servedio. Learning k-modal distributions via testing. In *SODA*, pages 1371–1385, 2012. 3.1

[DL96a] L. Devroye and G. Lugosi. Nonasymptotic universal smoothing factors, kernel complexity and Yatracos classes. *Annals of Statistics*, 25:2626–2637, 1996. 2.3

[DL96b] L. Devroye and G. Lugosi. A universally acceptable smoothing factor for kernel density estimation. *Annals of Statistics*, 24:2499–2512, 1996. 2.3

[DL01] L. Devroye and G. Lugosi. *Combinatorial methods in density estimation*. Springer Series in Statistics, Springer, 2001. 1.2, 1.3, 2.3, 2.3

[DP86] P. Deheuvels and D. Pfeifer. A semigroup approach to Poisson approximation. *Ann. Probab.*, 14:663–676, 1986. 1.2

[DP09] D. Dubhashi and A. Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, Cambridge, 2009. 1

[DP11] C. Daskalakis and C. Papadimitriou. On Oblivious PTAS's for Nash Equilibrium. *STOC* 2009, pp. 75–84. Full version available as ArXiV report, 2011. 1.3, 2

[DP13] C. Daskalakis and C. Papadimitriou. Sparse Covers for Sums of Indicators. *Arxiv Report*, 2013. http://arxiv.org/abs/1306.1265. 1.4, 2, 2, A, A

[Ehm91] Werner Ehm. Binomial approximation to the Poisson binomial distribution. *Statistics and Probability Letters*, 11:7–16, 1991. 1.2

[Ess42] Carl-Gustav Esseen. On the Liapunoff limit of error in the theory of probability. *Arkiv för matematik, astronomi och fysik*, A:1–19, 1942. 1.2

[Fil92] Sandra Fillebrown. Faster computation of Bernoulli numbers. *Journal of Algorithms*, 13(3):431 – 445, 1992. 6

[HC60] S.L. Hodges and L. Le Cam. The Poisson approximation to the binomial distribution. *Ann. Math. Statist.*, 31:747–740, 1960. 1.2

[Hoe63] W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963. 1

[Joh03] J.L. Johnson. *Probability and Statistics for Computer Science*. John Wiley & Sons, Inc., New York, NY, USA, 2003. 2.2

[KG71] J. Keilson and H. Gerber. Some Results for Discrete Unimodality. *J. American Statistical Association*, 66(334):386–389, 1971. 1.2, 2.1

[KMR$^+$94] 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*, pages 273–282, 1994. 1, 3

[KMV10] Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two Gaussians. In *STOC*, pages 553–562, 2010. 1.2

[Knu81] Donald E. Knuth. *The Art of Computer Programming, Volume II: Seminumerical Algorithms, 2nd Edition*. Addison-Wesley, 1981. 6

[Mik93] V.G. Mikhailov. On a refinement of the central limit theorem for sums of independent random indicators. *Theory Probab. Appl.*, 38:479–489, 1993. 1.2

[MV10] Ankur Moitra and Gregory Valiant. Settling the Polynomial Learnability of Mixtures of Gaussians. In *FOCS*, pages 93–102, 2010. 1.2

[NJ05] S. Kotz N.L. Johnson, A.W. Kemp. *Univariate discrete distributions*. John Wiley & Sons, Inc., New York, NY, USA, 2005. 2.2

[Poi37] S.D. Poisson. *Recherches sur la Probabilitè des jugements en matié criminelle et en matiére civile*. Bachelier, Paris, 1837. (document), 1

[PW11] Y. Peres and S. Watson. Personal communication, 2011. 2

[RÖ7] A. Röllin. Translated Poisson Approximation Using Exchangeable Pair Couplings. *Annals of Applied Probability*, 17(5/6):1596–1614, 2007. 1.3, 1, 1

[Roo00] B. Roos. Binomial approximation to the Poisson binomial distribution: The Krawtchouk expansion. *Theory Probab. Appl.*, 45:328–344, 2000. 1.2

[Sal76] Eugene Salamin. Computation of pi using arithmetic-geometric mean. *Mathematics of Computation*, 30(135):565–570, 1976. 3

[Sat93] Ken-Iti Sato. Convolution of unimodal distributions can produce any number of modes. *Annals of Probability*, 21(3):1543–1549, 1993. 3.1

[Soo96] S.Y.T. Soon. Binomial approximation for dependent indicators. *Statist. Sinica*, 6:703–714, 1996. 1.2

[SS71] A. Schönhage and V. Strassen. Schnelle multiplikation grosser zahlen. *Computing*, 7:281–292, 1971. 3

[Ste94] J.M. Steele. Le Cam's Inequality and Poisson Approximation. *Amer. Math. Monthly*, 101:48–54, 1994. 1.2

[Vol95] A. Yu. Volkova. A refinement of the central limit theorem for sums of independent random indicators. *Theory Probab. Appl.*, 40:791–794, 1995. 1.2

[VV11] Gregory Valiant and Paul Valiant. Estimating the unseen: an $n/\log(n)$-sample estimator for entropy and support size, shown optimal via new CLTs. In *STOC*, pages 685–694, 2011. 1.2

[Wan93] Y.H. Wang. On the number of successes in independent trials. *Statistica Sinica*, 3:295–312, 1993. 1.2

[Whi80] E.T. Whittaker. *A course of modern analysis*. Cambridge University Press, 1980. 1

[Yat85] Y. G. Yatracos. Rates of convergence of minimum distance estimators and Kolmogorov's entropy. *Annals of Statistics*, 13:768–774, 1985. 2.3

A Extension of the Cover Theorem: Proof of Theorem 4

Theorem 4 is restating the main cover theorem (Theorem 1) of [DP13], except that it claims an additional property, namely what follows the word "finally" in the statement of the theorem. (We will sometimes refer to this property as the *last part* of Theorem 4 in the following discussion.) Our goal is to show that the cover of [DP13] already satisfies this property without any modifications, thereby establishing Theorem 4. To avoid reproducing the involved constructions of [DP13], we will assume that the reader has some familiarity with them. Still, our proof here will be self-contained.

First, we note that the ϵ-cover $\mathcal{S}_\epsilon$ of Theorem 1 of [DP13] is a subset of a larger $\frac{\epsilon}{2}$-cover $\mathcal{S}'_{\epsilon/2}$ of size $n^2 + n \cdot (1/\epsilon)^{O(1/\epsilon^2)}$, which includes all the k-sparse and all the k-heavy Binomial PBDs (up to permutations of the underlying p_i's), for some $k = O(1/\epsilon)$. Let us call $\mathcal{S}'_{\epsilon/2}$ the "large $\frac{\epsilon}{2}$-cover" to distinguish it from $\mathcal{S}_\epsilon$, which we will call the "small ϵ-cover." The reader is referred to Theorem 2 in [DP13] (and the discussion following that theorem) for a description of the large $\frac{\epsilon}{2}$-cover, and to Section 3.2 of [DP13] for how this cover is used to construct the small ϵ-cover. In particular, the small ϵ-cover is a subset of the large $\epsilon/2$-cover, including only a subset of the sparse form distributions in the large $\epsilon/2$-cover. Moreover, for every sparse form distribution in the large $\epsilon/2$-cover, the small ϵ-cover includes at least one sparse form distribution that is $\epsilon/2$-close in total variation distance. Hence, if the large $\epsilon/2$-cover satisfies the last part of Theorem 4 (with $\epsilon/2$ instead of ϵ and $\mathcal{S}'_{\epsilon/2}$ instead of $\mathcal{S}_\epsilon$), it follows that the small ϵ-cover also satisfies the last part of Theorem 4.

So we proceed to argue that, for all ϵ, the large ϵ-cover implied by Theorem 2 of [DP13] satisfies the last part of Theorem 4. Let us first review how the large cover is constructed. (See Section 4 of [DP13] for the details.) For every collection of indicators $\{X_i\}_{i=1}^n$ with expectations $\{\mathbf{E}[X_i] = p_i\}_i$, the collection is subjected to two filters, called the *Stage 1* and *Stage 2* filters, and described respectively in Sections 4.1 and 4.2 of [DP13]. Using the same notation as [DP13], let us denote by $\{Z_i\}_i$ the collection output by the Stage 1 filter and by $\{Y_i\}_i$ the collection output by the Stage 2 filter. The collection $\{Y_i\}_i$ output by the Stage 2 filter satisfies $d_{\rm TV}(\sum_i X_i, \sum_i Y_i) \leq \epsilon$, and is included in the cover (possibly after permuting the Y_i's). Moreover, it is in sparse or heavy Binomial form. This way, it is made sure that, for every $\{X_i\}_i$, there exists some $\{Y_i\}_i$ in the cover that is ϵ-close and is in sparse or heavy Binomial form. We proceed to show that the cover thus defined satisfies the last part of Theorem 4.

For $\{X_i\}_i$, $\{Y_i\}_i$ and $\{Z_i\}_i$ as above, let (μ, σ^2), (μ_Z, σ_Z^2) and (μ_Y, σ_Y^2) denote respectively the (mean, variance) pairs of the variables $X = \sum_i X_i$, $Z = \sum_i Z_i$ and $Y = \sum_i Y_i$. We argue first that the pair (μ_Z, σ_Z^2) satisfies $|\mu - \mu_Z| = O(\epsilon)$ and $|\sigma^2 - \sigma_Z^2| = O(\epsilon \cdot (1 + \sigma^2))$. Next we argue that, if the collection $\{Y_i\}_i$ output by the Stage 2 filter is in heavy Binomial form, then (μ_Y, σ_Y^2) satisfies $|\mu - \mu_Y| = O(1)$ and $|\sigma^2 - \sigma_Y^2| = O(1 + \epsilon \cdot (1 + \sigma^2))$, concluding the proof.

- Proof for (μ_Z, σ_Z^2): The Stage 1 filter only modifies the indicators X_i with $p_i \in (0, 1/k) \cup (1-1/k, 1)$, for some well-chosen $k = O(1/\epsilon)$. For convenience let us define $\mathcal{L}_k = \{i \,|\, p_i \in (0, 1/k)\}$ and $\mathcal{H}_k = \{i \,|\, p_i \in (1 - 1/k, 1)\}$ as in [DP13]. The filter of Stage 1 rounds the expectations of the indicators indexed by $\mathcal{L}_k$ to some value in $\{0, 1/k\}$ so that no single expectation is altered by more than an additive $1/k$, and the sum of these expectations is not modified by more than an additive $1/k$. Similarly, the expectations of the indicators indexed by $\mathcal{H}_k$ are rounded to some value in $\{1 - 1/k, 1\}$. See the details of how the rounding is performed in Section 4.1 of [DP13]. Let us then denote by $\{p_i'\}_i$ the expectations of the indicators $\{Z_i\}_i$ resulting from the rounding. We argue that the mean and variance of $Z = \sum_i Z_i$ is close to the mean and variance of X. Indeed,

$$\begin{aligned} |\mu - \mu_Z| &= \left|\sum_i p_i - \sum_i p_i'\right| \\ &= \left|\sum_{i \in \mathcal{L}_k \cup \mathcal{H}_k} p_i - \sum_{i \in \mathcal{L}_k \cup \mathcal{H}_k} p_i'\right| \\ &\leq O(1/k) = O(\epsilon). \end{aligned} \tag{11}$$

Similarly,

$$\begin{aligned} |\sigma^2 - \sigma_Z^2| &= \left|\sum_i p_i(1-p_i) - \sum_i p_i'(1-p_i')\right| \\ &\leq \left|\sum_{i \in \mathcal{L}_k} p_i(1-p_i) - \sum_{i \in \mathcal{L}_k} p_i'(1-p_i')\right| + \left|\sum_{i \in \mathcal{H}_k} p_i(1-p_i) - \sum_{i \in \mathcal{H}_k} p_i'(1-p_i')\right|. \end{aligned}$$

We proceed to bound the two terms of the RHS separately. Since the argument is symmetric for $\mathcal{L}_k$ and

$\mathcal{H}_k$ we only do $\mathcal{L}_k$. We have

$$\begin{aligned}
\left|\sum_{i\in\mathcal{L}_k} p_i(1-p_i) - \sum_{i\in\mathcal{L}_k} p'_i(1-p'_i)\right| &= \left|\sum_{i\in\mathcal{L}_k}(p_i-p'_i)(1-(p_i+p'_i))\right| \\
&= \left|\sum_{i\in\mathcal{L}_k}(p_i-p'_i) - \sum_{i\in\mathcal{L}_k}(p_i-p'_i)(p_i+p'_i)\right| \\
&\le \left|\sum_{i\in\mathcal{L}_k}(p_i-p'_i)\right| + \left|\sum_{i\in\mathcal{L}_k}(p_i-p'_i)(p_i+p'_i)\right| \\
&\le \frac{1}{k} + \sum_{i\in\mathcal{L}_k}|p_i-p'_i|(p_i+p'_i) \\
&\le \frac{1}{k} + \frac{1}{k}\sum_{i\in\mathcal{L}_k}(p_i+p'_i) \\
&\le \frac{1}{k} + \frac{1}{k}\left(2\sum_{i\in\mathcal{L}_k}p_i + 1/k\right) \\
&= \frac{1}{k} + \frac{1}{k}\left(\frac{2}{1-1/k}\sum_{i\in\mathcal{L}_k}p_i(1-1/k) + 1/k\right) \\
&\le \frac{1}{k} + \frac{1}{k}\left(\frac{2}{1-1/k}\sum_{i\in\mathcal{L}_k}p_i(1-p_i) + 1/k\right) \\
&\le \frac{1}{k} + \frac{1}{k^2} + \frac{2}{k-1}\sum_{i\in\mathcal{L}_k}p_i(1-p_i).
\end{aligned}$$

Using the above (and a symmetric argument for index set $\mathcal{H}_k$) we obtain:

$$|\sigma^2 - \sigma_Z^2| \le \frac{2}{k} + \frac{2}{k^2} + \frac{2}{k-1}\sigma^2 = O(\epsilon)(1+\sigma^2). \tag{12}$$

- Proof for (μ_Y, σ_Y^2): After the Stage 1 filter is applied to the collection $\{X_i\}_i$, the resulting collection of random variables $\{Z_i\}_i$ has expectations $p'_i \in \{0,1\} \cup [1/k, 1-1/k]$, for all i. The Stage 2 filter has different form depending on the cardinality of the set $\mathcal{M} = \{i \mid p'_i \in [1/k, 1-1/k]\}$. In particular, if $|\mathcal{M}| > k^3$ the output of the Stage 2 filter is in heavy Binomial form, while if $|\mathcal{M}| \le k^3$ the output of the Stage 2 filter is in sparse form. As we are only looking to provide guarantee for the distributions in heavy Binomial form, it suffices to only consider the former case next.

  - $|\mathcal{M}| > k^3$: Let $\{Y_i\}_i$ be the collection produced by Stage 2 and let $Y = \sum_i Y_i$. Then Lemma 4 of [DP13] implies that

    $$|\mu_Z - \mu_Y| = O(1) \text{ and } |\sigma_Z^2 - \sigma_Y^2| = O(1).$$

    Combining this with (11) and (12) gives

    $$|\mu - \mu_Y| = O(1) \text{ and } |\sigma^2 - \sigma_Y^2| = O(1 + \epsilon\cdot(1+\sigma^2)).$$

This concludes the proof of Theorem 4. □

B Birgé's theorem: Learning unimodal distributions

Here we briefly explain how Theorem 5 follows from [Bir97]. We assume that the reader is moderately familiar with the paper [Bir97].

Birgé (see his Theorem 1 and Corollary 1) upper bounds the expected variation distance between the target distribution (which he denotes f) and the hypothesis distribution that is constructed by his algorithm (which he denotes $\hat{f}_n$; it should be noted, though, that his "n" parameter denotes the number of samples used by the algorithm, while we will denote this by "m", reserving "n" for the domain $\{1, \ldots, n\}$ of the distribution). More precisely, [Bir97] shows that this expected variation distance is at most that of the Grenander estimator (applied to learn a unimodal distribution when the mode is known) plus a lower-order term. For our Theorem 5 we take Birgé's "η" parameter to be ϵ. With this choice of η, by the results of [Bir87a, Bir87b] bounding the expected error of the Grenander estimator, if $m = O(\log(n)/\epsilon^3)$ samples are used in Birgé's algorithm then the expected variation distance between the target distribution and his hypothesis distribution is at most $O(\epsilon)$. To go from expected error $O(\epsilon)$ to an $O(\epsilon)$-accurate hypothesis with probability at least $1 - \delta$, we run the above-described algorithm $O(\log(1/\delta))$ times so that with probability at least $1 - \delta$ some hypothesis obtained is $O(\epsilon)$-accurate. Then we use our hypothesis testing procedure of Lemma 8, or, more precisely, the extension provided in Lemma 10, to identify an $O(\epsilon)$-accurate hypothesis from within this pool of $O(\log(1/\delta))$ hypotheses. (The use of Lemma 10 is why the running time of Theorem 5 depends quadratically on $\log(1/\delta)$ and why the sample complexity contains the second $\frac{1}{\epsilon^2} \log \frac{1}{\delta} \log \log \frac{1}{\delta}$ term.)

It remains only to argue that a single run of Birgé's algorithm on a sample of size $m = O(\log(n)/\epsilon^3)$ can be carried out in $\tilde{O}(\log^2(n)/\epsilon^3)$ bit operations (recall that each sample is a $\log(n)$-bit string). His algorithm begins by locating an $r \in [n]$ that approximately minimizes the value of his function $d(r)$ (see Section 3 of [Bir97]) to within an additive $\eta = \epsilon$ (see Definition 3 of his paper); intuitively this r represents his algorithm's "guess" at the true mode of the distribution. To locate such an r, following Birgé's suggestion in Section 3 of his paper, we begin by identifying two consecutive points in the sample such that r lies between those two sample points. This can be done using $\log m$ stages of binary search over the (sorted) points in the sample, where at each stage of the binary search we compute the two functions d^- and d^+ and proceed in the appropriate direction. To compute the function $d^-(j)$ at a given point j (the computation of d^+ is analogous), we recall that $d^-(j)$ is defined as the maximum difference over $[1, j]$ between the empirical cdf and its convex minorant over $[1, j]$. The convex minorant of the empirical cdf (over m points) can be computed in $\tilde{O}((\log n)m)$ bit-operations (where the $\log n$ comes from the fact that each sample point is an element of $[n]$), and then by enumerating over all points in the sample that lie in $[1, j]$ (in time $O((\log n)m)$) we can compute $d^-(j)$. Thus it is possible to identify two adjacent points in the sample such that r lies between them in time $\tilde{O}((\log n)m)$. Finally, as Birgé explains in the last paragraph of Section 3 of his paper, once two such points have been identified it is possible to again use binary search to find a point r in that interval where $d(r)$ is minimized to within an additive η. Since the maximum difference between d^- and d_+ can never exceed 1, at most $\log(1/\eta) = \log(1/\epsilon)$ stages of binary search are required here to find the desired r.

Finally, once the desired r has been obtained, it is straightforward to output the final hypothesis (which Birgé denotes $\hat{f}_n$). As explained in Definition 3, this hypothesis is the derivative of $\tilde{F}_n^r$, which is essentially the convex minorant of the empirical cdf to the left of r and the convex majorant of the empirical cdf to the right of r. As described above, given a value of r these convex majorants and minorants can be computed in $\tilde{O}((\log n)m)$ time, and the derivative is simply a collection of uniform distributions as claimed. This concludes our sketch of how Theorem 5 follows from [Bir97].

C Efficient Evaluation of the Poisson Distribution

In this section we provide an efficient algorithm to compute an additive approximation to the Poisson probability mass function. It seems that this should be a basic operation in numerical analysis, but we were not able to find it explicitly in the literature. Our main result for this section is the following.

Theorem 6. *There is an algorithm that, on input a rational number $\lambda > 0$, and integers $k \geq 0$ and $t > 0$, produces an estimate $\widehat{p_k}$ such that*

$$|\widehat{p_k} - p_k| \leq \frac{1}{t},$$

where $p_k = \frac{\lambda^k e^{-\lambda}}{k!}$ is the probability that the Poisson distribution of parameter λ assigns to integer k. The running time of the algorithm is $\tilde{O}(\langle t \rangle^3 + \langle k \rangle \cdot \langle t \rangle + \langle \lambda \rangle \cdot \langle t \rangle)$.

Proof. Clearly we cannot just compute $e^{-\lambda}$, λ^k and $k!$ separately, as this will take time exponential in the description complexity of k and λ. We follow instead an indirect approach. We start by rewriting the target probability as follows

$$p_k = e^{-\lambda + k\ln(\lambda) - \ln(k!)}.$$

Motivated by this formula, let

$$E_k := -\lambda + k\ln(\lambda) - \ln(k!).$$

Note that $E_k \leq 0$. Our goal is to approximate E_k to within high enough accuracy and then use this approximation to approximate p_k.

In particular, the main part of the argument involves an efficient algorithm to compute an approximation $\widehat{\widehat{E_k}}$ to E_k satisfying

$$\left|\widehat{\widehat{E_k}} - E_k\right| \leq \frac{1}{4t} \leq \frac{1}{2t} - \frac{1}{8t^2}. \tag{13}$$

This approximation will have bit complexity $\tilde{O}(\langle k \rangle + \langle \lambda \rangle + \langle t \rangle)$ and be computable in time $\tilde{O}(\langle k \rangle \cdot \langle t \rangle + \langle \lambda \rangle + \langle t \rangle^3)$.

We show that if we had such an approximation, then we would be able to complete the proof. For this, we claim that it suffices to approximate $e^{\widehat{\widehat{E_k}}}$ to within an additive error $\frac{1}{2t}$. Indeed, if $\widehat{p_k}$ were the result of this approximation, then we would have:

$$\begin{aligned}
\widehat{p_k} &\leq e^{\widehat{\widehat{E_k}}} + \frac{1}{2t} \\
&\leq e^{E_k + \frac{1}{2t} - \frac{1}{8t^2}} + \frac{1}{2t} \\
&\leq e^{E_k + \ln(1 + \frac{1}{2t})} + \frac{1}{2t} \\
&\leq e^{E_k}\left(1 + \frac{1}{2t}\right) + \frac{1}{2t} \leq p_k + \frac{1}{t};
\end{aligned}$$

and similarly

$$\begin{aligned}
\widehat{p_k} &\geq e^{\widehat{\widehat{E_k}}} - \frac{1}{2t} \\
&\geq e^{E_k - (\frac{1}{2t} - \frac{1}{8t^2})} - \frac{1}{2t} \\
&\geq e^{E_k - \ln(1 + \frac{1}{2t})} - \frac{1}{2t} \\
&\geq e^{E_k} \Big/ \left(1 + \frac{1}{2t}\right) - \frac{1}{2t} \\
&\geq e^{E_k}\left(1 - \frac{1}{2t}\right) - \frac{1}{2t} \geq p_k - \frac{1}{t}.
\end{aligned}$$

To approximate $e^{\widehat{\widehat{E_k}}}$ given $\widehat{\widehat{E_k}}$, we need the following lemma:

Lemma 14. *Let $\alpha \leq 0$ be a rational number. There is an algorithm that computes an estimate $\widehat{e^\alpha}$ such that*

$$\left|\widehat{e^\alpha} - e^\alpha\right| \leq \frac{1}{2t}$$

and has running time $\tilde{O}(\langle\alpha\rangle \cdot \langle t\rangle + \langle t\rangle^2)$.

Proof. Since $e^\alpha \in [0,1]$, the point of the additive grid $\{\frac{i}{4t}\}_{i=1}^{4t}$ closest to e^α achieves error at most $1/(4t)$. Equivalently, in a logarithmic scale, consider the grid $\{\ln\frac{i}{4t}\}_{i=1}^{4t}$ and let $j^* := \arg\min_j \left\{\left|\alpha - \ln(\frac{j}{4t})\right|\right\}$. Then, we have that

$$\left|\frac{j^*}{(4t)} - e^\alpha\right| \leq \frac{1}{4t}.$$

The idea of the algorithm is to approximately identify the point j^*, by computing approximations to the points of the logarithmic grid combined with a binary search procedure. Indeed, consider the "rounded" grid $\{\widehat{\ln\frac{i}{4t}}\}_{i=1}^{4t}$ where each $\widehat{\ln(\frac{i}{4t})}$ is an approximation to $\ln(\frac{i}{4t})$ that is accurate to within an additive $\frac{1}{16t}$. Notice that, for $i = 1, \ldots, 4t$:

$$\ln\left(\frac{i+1}{4t}\right) - \ln\left(\frac{i}{4t}\right) = \ln\left(1 + \frac{1}{i}\right) \geq \ln\left(1 + \frac{1}{4t}\right) > 1/8t.$$

Given that our approximations are accurate to within an additive $1/16t$, it follows that the rounded grid $\{\widehat{\ln\frac{i}{4t}}\}_{i=1}^{4t}$ is monotonic in i.

The algorithm does not construct the points of this grid explicitly, but adaptively as it needs them. In particular, it performs a binary search in the set $\{1, \ldots, 4t\}$ to find the point $i^* := \arg\min_i \left\{\left|\alpha - \widehat{\ln(\frac{i}{4t})}\right|\right\}$. In every iteration of the search, when the algorithm examines the point j, it needs to compute the approximation $g_j = \widehat{\ln(\frac{j}{4t})}$ and evaluate the distance $|\alpha - g_j|$. It is known that the logarithm of a number x with a binary fraction of L bits and an exponent of $o(L)$ bits can be computed to within a relative error $O(2^{-L})$ in time $\tilde{O}(L)$ [Bre75]. It follows from this that g_j has $O(\langle t\rangle)$ bits and can be computed in time $\tilde{O}(\langle t\rangle)$. The subtraction takes linear time, i.e., it uses $O(\langle\alpha\rangle + \langle t\rangle)$ bit operations. Therefore, each step of the binary search can be done in time $O(\langle\alpha\rangle) + \tilde{O}(\langle t\rangle)$ and thus the overall algorithm has $O(\langle\alpha\rangle \cdot \langle t\rangle) + \tilde{O}(\langle t\rangle^2)$ running time.

The algorithm outputs $\frac{i^*}{4t}$ as its final approximation to e^α. We argue next that the achieved error is at most an additive $\frac{1}{2t}$. Since the distance between two consecutive points of the grid $\{\ln\frac{i}{4t}\}_{i=1}^{4t}$ is more than $1/(8t)$ and our approximations are accurate to within an additive $1/16t$, a little thought reveals that $i^* \in \{j^*-1, j^*, j^*+1\}$. This implies that $\frac{i^*}{4t}$ is within an additive $1/2t$ of e^α as desired, and the proof of the lemma is complete. □

Given Lemma 14, we describe how we could approximate $e^{\widehat{\widehat{E_k}}}$ given $\widehat{\widehat{E_k}}$. Recall that we want to output an estimate $\widehat{p_k}$ such that $|\widehat{p_k} - e^{\widehat{\widehat{E_k}}}| \leq 1/(2t)$. We distinguish the following cases:

- If $\widehat{\widehat{E_k}} \geq 0$, we output $\widehat{p_k} := 1$. Indeed, given that $\left|\widehat{\widehat{E_k}} - E_k\right| \leq \frac{1}{4t}$ and $E_k \leq 0$, if $\widehat{\widehat{E_k}} \geq 0$ then $\widehat{\widehat{E_k}} \in [0, \frac{1}{4t}]$. Hence, because $t \geq 1$, $e^{\widehat{\widehat{E_k}}} \in [1, 1 + 1/2t]$, so 1 is within an additive $1/2t$ of the right answer.
- Otherwise, $\widehat{p_k}$ is defined to be the estimate obtained by applying Lemma 14 for $\alpha := \widehat{\widehat{E_k}}$. Given the bit complexity of $\widehat{\widehat{E_k}}$, the running time of this procedure will be $\tilde{O}(\langle k\rangle \cdot \langle t\rangle + \langle\lambda\rangle \cdot \langle t\rangle + \langle t\rangle^2)$.

Hence, the overall running time is $\tilde{O}(\langle k\rangle \cdot \langle t\rangle + \langle\lambda\rangle \cdot \langle t\rangle + \langle t\rangle^3)$.

In view of the above, we only need to show how to compute $\widehat{\widehat{E_k}}$. There are several steps to our approximation:

1. (Stirling's Asymptotic Approximation): Recall Stirling's asymptotic approximation (see e.g., [Whi80] p.193), which says that $\ln k!$ equals

$$k\ln(k) - k + (1/2)\cdot\ln(2\pi) + \sum_{j=2}^{m} \frac{B_j \cdot (-1)^j}{j(j-1)\cdot k^{j-1}} + O(1/k^m)$$

where B_k are the Bernoulli numbers. We define an approximation of $\ln k!$ as follows:

$$\widehat{\ln k!} := k\ln(k) - k + (1/2)\cdot\ln(2\pi) + \sum_{j=2}^{m_0} \frac{B_j \cdot (-1)^j}{j(j-1)\cdot k^{j-1}}$$

for $m_0 := O\left(\left\lceil \frac{\langle t\rangle}{\langle k\rangle} \right\rceil + 1\right)$.

2. (Definition of an approximate exponent $\widehat{E_k}$): Define $\widehat{E_k} := -\lambda + k\ln(\lambda) - \widehat{\ln(k!)}$. Given the above discussion, we can calculate the distance of $\widehat{E_k}$ to the true exponent E_k as follows:

$$|E_k - \widehat{E_k}| \le |\ln(k!) - \widehat{\ln(k!)}| \le O(1/k^{m_0}) \tag{14}$$

$$\le \frac{1}{10t}. \tag{15}$$

So we can focus our attention to approximating $\widehat{E_k}$. Note that $\widehat{E_k}$ is the sum of $m_0 + 2 = O(\frac{\log t}{\log k})$ terms. To approximate it within error $1/(10t)$, it suffices to approximate each summand within an additive error of $O(1/(t\cdot\log t))$. Indeed, we so approximate each summand and our final approximation $\widehat{\widehat{E_k}}$ will be the sum of these approximations. We proceed with the analysis:

3. (Estimating 2π): Since 2π shows up in the above expression, we should try to approximate it. It is known that the first ℓ digits of π can be computed exactly in time $O(\log\ell\cdot M(\ell))$, where $M(\ell)$ is the time to multiply two ℓ-bit integers [Sal76, Bre76]. For example, if we use the Schönhage-Strassen algorithm for multiplication [SS71], we get $M(\ell) = O(\ell\cdot\log\ell\cdot\log\log\ell)$. Hence, choosing $\ell := \lceil\log_2(12t\cdot\log t)\rceil$, we can obtain in time $\tilde{O}(\langle t\rangle)$ an approximation $\widehat{2\pi}$ of 2π that has a binary fraction of ℓ bits and satisfies:

$$|\widehat{2\pi} - 2\pi| \le 2^{-\ell} \;\;\Rightarrow\;\; (1-2^{-\ell})2\pi \le \widehat{2\pi} \le (1+2^{-\ell})2\pi.$$

Note that, with this approximation, we have

$$\left|\ln(2\pi) - \ln(\widehat{2\pi})\right| \le \ln(1-2^{-\ell}) \le 2^{-\ell} \le 1/(12t\cdot\log t).$$

4. (Floating-Point Representation): We will also need accurate approximations to $\ln\widehat{2\pi}$, $\ln k$ and $\ln\lambda$. We think of $\widehat{2\pi}$ and k as multiple-precision floating point numbers base 2. In particular,

   - $\widehat{2\pi}$ can be described with a binary fraction of $\ell + 3$ bits and a constant size exponent; and
   - $k \equiv 2^{\lceil\log k\rceil}\cdot\frac{k}{2^{\lceil\log k\rceil}}$ can be described with a binary fraction of $\lceil\log k\rceil$, i.e., $\langle k\rangle$, bits and an exponent of length $O(\log\log k)$, i.e., $O(\log\langle k\rangle)$.

   Also, since λ is a positive rational number, $\lambda = \frac{\lambda_1}{\lambda_2}$, where λ_1 and λ_2 are positive integers of at most $\langle\lambda\rangle$ bits. Hence, for $i = 1, 2$, we can think of λ_i as a multiple-precision floating point number base 2 with a binary fraction of $\langle\lambda\rangle$ bits and an exponent of length $O(\log\langle\lambda\rangle)$. Hence, if we choose $L = \lceil\log_2(12(3k+1)t^2\cdot k\cdot\lambda_1\cdot\lambda_2)\rceil = O(\langle k\rangle + \langle\lambda\rangle + \langle t\rangle)$, we can represent all numbers $\widehat{2\pi}, \lambda_1, \lambda_2, k$ as multiple precision floating point numbers with a binary fraction of L bits and an exponent of $O(\log L)$ bits.

5. (Estimating the logs): It is known that the logarithm of a number x with a binary fraction of L bits and an exponent of $o(L)$ bits can be computed to within a relative error $O(2^{-L})$ in time $\tilde{O}(L)$ [Bre75]. Hence, in time $\tilde{O}(L)$ we can obtain approximations $\widehat{\ln \widehat{2\pi}}, \widehat{\ln k}, \widehat{\ln \lambda_1}, \widehat{\ln \lambda_2}$ such that:
   - $|\widehat{\ln k} - \ln k| \leq 2^{-L} \ln k \leq \frac{1}{12(3k+1)t^2}$; and similarly
   - $|\widehat{\ln \lambda_i} - \ln \lambda_i| \leq \frac{1}{12(3k+1)t^2}$, for $i = 1, 2$;
   - $|\widehat{\ln \widehat{2\pi}} - \ln \widehat{2\pi}| \leq \frac{1}{12(3k+1)t^2}$.

6. (Estimating the terms of the series): To complete the analysis, we also need to approximate each term of the form $c_j = \frac{B_j}{j(j-1) \cdot k^{j-1}}$ up to an additive error of $O(1/(t \cdot \log t))$. We do this as follows: We compute the numbers B_j and k^{j-1} exactly, and we perform the division approximately.

   Clearly, the positive integer k^{j-1} has description complexity $j \cdot \langle k \rangle = O(m_0 \cdot \langle k \rangle) = O(\langle t \rangle + \langle k \rangle)$, since $j = O(m_0)$. We compute k^{j-1} exactly using repeated squaring in time $\tilde{O}(j \cdot \langle k \rangle) = \tilde{O}(\langle t \rangle + \langle k \rangle)$. It is known [Fil92] that the rational number B_j has $\tilde{O}(j)$ bits and can be computed in $\tilde{O}(j^2) = \tilde{O}(\langle t \rangle^2)$ time. Hence, the approximate evaluation of the term c_j (up to the desired additive error of $1/(t \log t)$) can be done in $\tilde{O}(\langle t \rangle^2 + \langle k \rangle)$, by a rational division operation (see e.g., [Knu81]). The sum of all the approximate terms takes linear time, hence the approximate evaluation of the entire truncated series (comprising at most $m_0 \leq \langle t \rangle$ terms) can be done in $\tilde{O}(\langle t \rangle^3 + \langle k \rangle \cdot \langle t \rangle)$ time overall.

   Let $\widehat{\widehat{E_k}}$ be the approximation arising if we use all the aforementioned approximations. It follows from the above computations that

$$\left| \widehat{\widehat{E_k}} - \widehat{E_k} \right| \leq \frac{1}{10t}.$$

7. (Overall Error): Combining the above computations we get:

$$\left| \widehat{\widehat{E_k}} - E_k \right| \leq \frac{1}{4t}.$$

   The overall time needed to obtain $\widehat{\widehat{E_k}}$ was $\tilde{O}(\langle k \rangle \cdot \langle t \rangle + \langle \lambda \rangle + \langle t \rangle^3)$ and the proof of Theorem 6 is complete. □

□