Learning mixtures of structured distributions over discrete domains

Siu-On Chan* Ilias Diakonikolas† Rocco A. Servedio‡ Xiaorui Sun§

July 3, 2012

Abstract

Let $\mathfrak{C}$ be a class of probability distributions over the discrete domain $[n] = \{1, \ldots, n\}$. We show that if $\mathfrak{C}$ satisfies a rather general condition – essentially, that each distribution in $\mathfrak{C}$ can be well-approximated by a variable-width histogram with few bins – then there is a highly efficient (both in terms of running time and sample complexity) algorithm that can learn any mixture of k unknown distributions from $\mathfrak{C}$.

We analyze several natural types of distributions over $[n]$, including log-concave, monotone hazard rate and unimodal distributions, and show that they have the required structural property of being well-approximated by a histogram with few bins. Applying our general algorithm, we obtain near-optimally efficient algorithms for all these mixture learning problems as described below. More precisely,

- **Log-concave distributions:** We learn any mixture of k log-concave distributions over $[n]$ using $k \cdot \tilde{O}(1/\varepsilon^4)$ samples (independent of n) and running in time $\tilde{O}(k \log(n)/\varepsilon^4)$ bit-operations (note that reading a single sample from $[n]$ takes $\Theta(\log n)$ bit operations). For the special case $k = 1$ we give an efficient algorithm using $\tilde{O}(1/\varepsilon^3)$ samples; this generalizes the main result of [DDS12b] from the class of Poisson Binomial Distributions to the much broader class of all log-concave distributions. Our upper bounds are not far from optimal since any algorithm for this learning problem requires $\Omega(k/\varepsilon^{5/2})$ samples.
- **Monotone hazard rate (MHR) distributions:** We learn any mixture of k MHR distributions over $[n]$ using $O(k \log(n/\varepsilon)/\varepsilon^4)$ samples and running in time $\tilde{O}(k \log^2(n)/\varepsilon^4)$ bit-operations. Any algorithm for this learning problem must use $\Omega(k \log(n)/\varepsilon^3)$ samples.
- **Unimodal distributions:** We give an algorithm that learns any mixture of k unimodal distributions over $[n]$ using $O(k \log(n)/\varepsilon^4)$ samples and running in time $\tilde{O}(k \log^2(n)/\varepsilon^4)$ bit-operations. Any algorithm for this problem must use $\Omega(k \log(n)/\varepsilon^3)$ samples.

*siuon@cs.berkeley.edu.

†ilias@cs.berkeley.edu. Research supported by a Simons Postdoctoral Fellowship.

‡rocco@cs.columbia.edu. Supported by NSF grants CCF-0915929 and CCF-1115703.

§xiaoruisun@cs.columbia.edu.

1 Introduction

1.1 Background and motivation. Learning an unknown probability distribution given access to independent samples is a classical topic with a long history in statistics and probability theory. Theoretical computer science researchers have also been interested in these problems at least since the 1990s [KMR$^+$94, Das99], with an explicit focus on the *computational efficiency* of algorithms for learning distributions. Many works in theoretical computer science over the past decade have focused on learning and testing various kinds of probability distributions over high-dimensional spaces, see e.g. [Das99, FM99, DS00, AK01, VW02, FOS05, RS05, BS10, KMV10, MV10, ACS10] and references therein. There has also been significant recent interest in learning and testing various types of probability distributions over the discrete domain $[n] = \{1, \ldots, n\}$, see e.g. [BKR04, VV11b, VV11a, DDS12a, DDS12b].

A natural type of distribution learning problem, which is the focus of this work, is that of learning a unknown *mixture* of "simple" distributions. Mixtures of distributions have received much attention in statistics [Lin95, RW84, TSM85] and in recent years have been intensively studied in computer science as well (see many of the papers referenced above). Given distributions $p_1, \ldots, p_k$ and non-negative values $\mu_1, \ldots, \mu_k$ that sum to 1, we say that $p = \sum_{i=1}^{k} \mu_i p_i$ is a *k-mixture* of *components* $p_1, \ldots, p_k$ with *mixing weights* $\mu_1, \ldots, \mu_k$. A draw from p is obtained by choosing $i \in [k]$ with probability μ_i and then making a draw from p_i.

In this paper we work in essentially the classical "density estimation" framework [Sil86, Sco92, DL01] which is very similar to the model considered in [KMR$^+$94] in a theoretical computer science context. In this framework the learning algorithm is given access to independent samples drawn from an unknown target distribution over $[n]$, and it must output a hypothesis distribution h over $[n]$ such that with high probability the total variation distance $d_{\mathrm{TV}}(p, h)$ between p and h is at most ε. Thus, for learning mixture distributions, our goal is simply to construct a high-accuracy hypothesis distribution which is very close to the mixture distribution that generated the data. In keeping with the spirit of [KMR$^+$94], we shall be centrally concerned with the *running time* as well as the number of samples required by our algorithms that learn mixtures of various types of discrete distributions over $[n]$.

We focus on *density estimation* rather than, say, clustering or parameter estimation, for several reasons. First, clustering samples according to which component in the mixture each sample came from is often an impossible task unless restrictive separation assumptions are made on the components; we prefer not to make such assumptions. Second, the classes of distributions that we are chiefly interested in (such as log-concave, MHR and unimodal distributions) are all non-parametric classes, so it is unclear what "parameter estimation" would even mean for these classes. Finally, even in highly restricted special cases, parameter estimation provably requires sample complexity *exponential* in k, the number of components in the mixture. Moitra and Valiant [MV10] have shown that parameter estimation for a mixture of k Gaussians inherently requires $\exp(\Omega(k)))$ samples. Their argument can be translated to the discrete setting, with translated Binomial distributions in place of Gaussians, to provide a similar lower bound for parameter estimation of translated Binomial mixtures. Thus, parameter estimation even for a mixture of k translated Binomial distributions over $[n]$ (a highly restricted special case of *all* the mixture classes we consider, since translated Binomial distributions are log-concave, MHR and unimodal) requires $\exp(\Omega(k))$ samples. This rather discouraging lower bound motivates the study of other variants of the problem of learning mixture distributions.

Returning to our density estimation framework, it is not hard to show that from an information-theoretic perspective, learning a mixture of distributions from a class $\mathfrak{C}$ is never much harder than learning a single distribution from $\mathfrak{C}$. In Appendix A we give a simple argument which establishes the following:

Proposition 1.1. *[Sample Complexity of Learning Mixtures] Let $\mathfrak{C}$ be a class of distributions over $[n]$. Let A be an algorithm which learns any unknown distribution p in $\mathfrak{C}$ using $m(n, \varepsilon) \cdot \log(1/\delta)$ samples, i.e. with probability $1 - \delta$ A outputs a hypothesis h such that $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Then there is an algorithm A' which uses $\tilde{O}(k/\varepsilon^3) \cdot m(n, \varepsilon/20) \cdot \log^2(1/\delta)$ samples and learns any unknown k-mixture of distributions in $\mathfrak{C}$ to*

variation distance ε with confidence $1-\delta$.

While the generic algorithm A' uses relatively few samples, it is computationally highly inefficient, with running time exponentially higher than the runtime of algorithm A (since A' tries all possible partitions of its input sample into k separate subsamples). Indeed, naive approaches to learning mixture distributions run into a "credit assignment" problem of determining which component distribution each sample point belongs to.

As the main contributions of this paper, we (i) give a general algorithm which *efficiently* learns mixture distributions over $[n]$ provided that the component distributions satisfy a mild condition; and (ii) show that this algorithm can be used to obtain highly efficient algorithms for natural mixture learning problems.

1.2 A general algorithm. The mild condition which we require of the component distributions in our mixtures is essentially that each component distribution must be close to a (variable-width) histogram with few bins. More precisely, let us say that a distribution q over $[n]$ is (ε, t)-*flat* (see Section 2) if there is a partition of $[n]$ into t disjoint intervals $I_1, \ldots, I_t$ such that p is ε-close (in total variation distance) to the distribution obtained by "flattening" p within each interval I_j (i.e. by replacing $p(k)$, for $k \in I_j$, with $\sum_{i \in I_j} p(i)/|I_j|$). Our general result for learning mixture distributions is a highly efficient algorithm that learns any k-mixture of (ε, t)-flat distributions:

Theorem 1.1 (informal statement)**.** *There is an algorithm that learns any k-mixture of (ε, t)-flat distributions over $[n]$ to accuracy $O(\varepsilon)$, using $O(kt/\varepsilon^3)$ samples and running in $\tilde{O}(kt \log(n)/\varepsilon^3)$ bit-operations.*

As we show in Section 1.3 below, Theorem 1.1 yields near-optimal sample complexity for a range of interesting mixture learning problems, with a running time that is nearly linear in the sample size. Another attractive feature of Theorem 1.1 is that it always outputs hypothesis distributions with a very simple structure (enabling a succinct representation), namely histograms with at most kt/ε bins.

1.3 Applications of the general approach. We apply our general approach to obtain a wide range of learning results for mixtures of various natural and well-studied types of discrete distributions. These include mixtures of *log-concave* distributions, mixtures of *monotone hazard rate (MHR)* distributions, and mixtures of *unimodal* distributions. To do this, in each case we need a structural result stating that any distribution of the relevant type can be well-approximated by a histogram with few bins. In some cases (unimodal distributions) the necessary structural results were previously known, but in others (log-concave and MHR distributions) we establish novel structural results that, combined with our general approach, yield nearly optimal algorithms.

Log-concave distributions. Discrete log-concave distributions are essentially those distributions p that satisfy $p(k)^2 \geq p(k+1)p(k-1)$ (see Section 4 for a precise definition). They are closely analogous to log-concave distributions over continuous domains, and encompass a range of interesting and well-studied types of discrete distributions, including binomial, negative binomial, geometric, hypergeometric, Poisson, Poisson Binomial, hyper-Poisson, Pólya-Eggenberger, and Skellam distributions (see Section 1 of [FBR11]). In the continuous setting, log-concave distributions include include uniform, normal, exponential, logistic, extreme value, Laplace, Weibull, Gamma, Chi and Chi-Squared and Beta distributions, see [BB05]. Log-concave distributions over $[n]$ have been studied in a range of different contexts including economics, statistics and probability theory, and algebra, combinatorics and geometry, see [An95, FBR11, Sta89] and references therein.

Our main learning result for mixtures of discrete log-concave distributions is:

Theorem 1.2. *There is an algorithm that learns any k-mixture of log-concave distributions over $[n]$ to variation distance ε using $k \cdot \tilde{O}(1/\varepsilon^4)$ samples and running in $\tilde{O}(k \log(n)/\varepsilon^4)$ bit-operations.*

We stress that the sample complexity above is completely independent of the domain size n. In the special case of learning a single discrete log-concave distribution we achieve an improved sample complexity of $\tilde{O}(1/\varepsilon^3)$ samples, with running time $\tilde{O}(\log(n)/\varepsilon^3)$. This matches the sample complexity and running time of the main result of [DDS12b], which was a specialized algorithm for learning Poisson Binomial Distribution over $[n]$. Our new algorithm is simpler, applies to the broader class of all log-concave distributions, has a much simpler and more self-contained analysis, and generalizes to mixtures of k distributions (at the cost of an additional $1/\varepsilon$ factor in runtime and sample complexity). We note that these algorithmic results are not far from the best possible for mixtures of log-concave distributions. We show in Section 4 that for $k \leq n^{1-\Omega(1)}$ and $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a mixture of k log-concave distributions to accuracy ε must use $\Omega(k/\varepsilon^{2.5})$ samples.

Monotone Hazard Rate (MHR) distributions. A discrete distribution p over $[n]$ is said to have a *monotone (increasing) hazard rate* if the *hazard rate* $H(i) \stackrel{\text{def}}{=} \frac{p(i)}{\sum_{j \geq i} p(j)}$ is a non-decreasing function of i. It is well known that every discrete log-concave distribution is MHR (see e.g. part (ii) of Proposition 10 of [An95]), but MHR is a more general condition than log-concavity (for example, it is easy to check that every non-decreasing distribution over $[n]$ is MHR, but such distributions need not be log-concave). MHR distributions arise frequently in mechanism design [Mye81] and reliability theory; [BMP63] is a good reference for basic properties of these distributions.

Our main learning result for mixtures of MHR distributions is:

Theorem 1.3. *There is an algorithm that learns any k-mixture of MHR distributions over $[n]$ to variation distance ε using $O(k \log(n/\varepsilon)/\varepsilon^4)$ samples and running in $\tilde{O}(k \log^2(n)/\varepsilon^4)$ bit-operations.*

This theorem is also nearly optimal. We show that for $k \leq n^{1-\Omega(1)}$ and $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a mixture of k MHR distributions to accuracy ε must use $\Omega(k \log(n)/\varepsilon^3)$ samples.

Unimodal distributions. A distribution over $[n]$ is said to be *unimodal* if its pdf is monotone non-decreasing over $[1, t]$ for some $t \leq n$ and then monotone non-increasing on $[t, n]$. Every log-concave distribution is unimodal, but the MHR and unimodal conditions are easily seen to be incomparable. Many natural types of distributions are unimodal and there has been extensive work on density estimation for unimodal distributions and related questions [Rao69, Weg70, BKR04, Bir97, Fou97].

Our main learning result for mixtures of unimodal distributions is:

Theorem 1.4. *There is an algorithm that learns any k-mixture of unimodal distributions over $[n]$ to variation distance ε using $O(k \log(n)/\varepsilon^4)$ samples and running in $\tilde{O}(k \log^2(n)/\varepsilon^4)$ bit-operations.*

Our approach in fact extends to learning a k-mixture of t-modal distributions (see Appendix D). The same lower bound argument that we use for mixtures of MHR distributions also gives us that for $k \leq n^{1-\Omega(1)}$ and $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a mixture of k unimodal distributions to accuracy ε must use $\Omega(k \log(n)/\varepsilon^3)$ samples.

1.4 Related work. Log-concave distributions: Maximum likelihood estimators for both continuous [DR09, Wal09] and discrete [FBR11] log-concave distributions have been recently studied by various authors. For special cases of log-concave densities over $\mathbb{R}$ (that satisfy various restrictions on the shape of the pdf) upper bounds on the minimax risk of estimators are known, see e.g. Exercise 15.21 of [DL01]. (We remark that these results do not imply the $k = 1$ case of our log-concave mixture learning result.) Perhaps the most relevant prior work is the recent algorithm of [DDS12b] which gives a $\tilde{O}(1/\varepsilon^3)$-sample, $\tilde{O}(\log(n)/\varepsilon^3)$-time algorithm for learning any Poisson Binomial Distribution over $[n]$. (As noted above, we match the performance of the [DDS12b] algorithm for the broader class of all log-concave distributions, as the $k = 1$ case of our log-concave mixture learning result.)

Achlioptas and McSherry [AM05] and Kannan et al. [KSV08] gave algorithms for clustering points drawn from a mixture of k high-dimensional log-concave distributions, under various separation assumptions on the distance between the means of the components. We are not aware of prior work on density estimation of mixtures of arbitrary log-concave distributions in either the continuous or the discrete setting.

MHR distributions: As noted above, MHR distributions appear frequently and play an important role in reliability theory and in economics (to the extent that the MHR condition is considered a standard assumption in these settings). Surprisingly, the problem of learning an unknown MHR distribution or mixture of such distributions has not been explicitly considered in the statistics literature. We note that several authors have considered the problem of estimating the hazard rate of an MHR distribution in different contexts, see e.g. [Wan86, HW93, GJ11, Ban08].

Unimodal distributions: The problem of learning a single unimodal distribution is well-understood: Birgé [Bir97] gave an efficient algorithm for learning continuous unimodal distributions (whose density is absolutely bounded); his algorithm, when translated to the discrete domain $[n]$, requires $O(\log(n)/\varepsilon^3)$ samples. This sample size is also known to be optimal (up to constant factors)[Bir87a]. In recent work, Daskalakis et al. [DDS12a] gave an efficient algorithm to learn t-modal distributions over $[n]$. We remark that their result does not imply ours, as even a mixture of two unimodal distributions over $[n]$ may have $\Omega(n)$ modes. We are not aware of prior work on efficiently learning mixtures of unimodal distributions.

Paper Structure. Following some preliminaries in Section 2, Section 3 presents our general framework for learning mixtures. Sections 4 and 5 analyze the cases of log-concave and MHR mixtures respectively. Due to space limitations, the case of unimodal mixtures, as well as most proofs are deferred to an Appendix.

2 Preliminaries and notation

We write $[n]$ to denote the discrete domain $\{1, \ldots, n\}$ and $[i, j]$ to denote the set $\{i, \ldots, j\}$ for $i \leq j$. For $v = (v(1), \ldots, v(n)) \in \mathbb{R}^n$ we write $\|v\|_1 = \sum_{i=1}^n |v(i)|$ to denote its L_1-norm.

For p a probability distribution over $[n]$ we write $p(i)$ to denote the probability of element $i \in [n]$ under p, so $p(i) \geq 0$ for all $i \in [n]$ and $\sum_{i=1}^n p(i) = 1$. For $S \subseteq [n]$ we write $p(S)$ to denote $\sum_{i \in S} p(i)$. We write p^S to denote the *sub-distribution* over S induced by p, i.e. $p^S(i) = p(i)$ if $i \in S$ and $p^S(i) = 0$ otherwise.

A distribution p over $[n]$ is non-increasing (resp. non-decreasing) if $p(i+1) \leq p(i)$ (resp. $p(i+1) \geq p(i)$), for all $i \in [n-1]$; p is *monotone* if it is either non-increasing or non-decreasing.

Let p, q be distributions over $[n]$ with corresponding cdfs P, Q. The *total variation distance* between p and q is $d_{\mathrm{TV}}(p, q) \stackrel{\text{def}}{=} \max_{S \subseteq [n]} |p(S) - q(S)| = (1/2) \cdot \|p - q\|_1$. The *Kolmogorov distance* between p and q is defined as $d_{\mathrm{K}}(p, q) \stackrel{\text{def}}{=} \max_{j \in [n]} |\sum_{i=1}^j p(i) - \sum_{i=1}^j q(i)|$. Note that $d_{\mathrm{K}}(p, q) \leq d_{\mathrm{TV}}(p, q)$.

Finally, the following notation and terminology will be useful: given m independent samples $s_1, \ldots, s_m$, drawn from distribution $p : [n] \to [0, 1]$, the *empirical distribution* $\widehat{p}_m : [n] \to [0, 1]$ is defined as follows: for all $i \in [n]$, $\widehat{p}_m(i) = |\{j \in [m] \mid s_j = i\}|/m$.

Partitions, flat decompositions and refinements. Given a partition $\mathcal{I} = \{I_1, \ldots, I_t\}$ of $[n]$ into t disjoint intervals and a distribution p over $[n]$, we write $p^{\mathrm{flat}(\mathcal{I})}$ to denote the *flattened distribution*. This is the distribution over $[n]$ defined as follows: for $j \in [t]$ and $i \in I_j$, $p^{\mathrm{flat}(\mathcal{I})}(i) = p(I_j)/|I_j|$. That is, $p^{\mathrm{flat}(\mathcal{I})}$ is obtained from p by averaging the weight that p assigns to each interval in $\mathcal{I}$ over the entire interval.

Definition 2.1 (Flat decomposition)**.** *Let p be a distribution over $[n]$ and $\mathcal{P}$ be a partition of $[n]$ into t disjoint intervals. We say that $\mathcal{P}$ is a (p, ε, t)-*flat decomposition of $[n]$ *if $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) \leq \varepsilon$. If there exists a (p, ε, t)-flat decomposition of $[n]$ then we say that p is (ε, t)-*flat.

Let $\mathcal{I} = \{I_1, \ldots, I_s\}$ be a partition of $[n]$ into s disjoint intervals, and $\mathcal{J} = \{J_1, \ldots, J_t\}$ be a partition of

$[n]$ into t disjoint intervals. We say that $\mathcal{J}$ is a *refinement* of $\mathcal{I}$ if each interval in $\mathcal{I}$ is a union of intervals in $\mathcal{J}$, i.e. for every $a \in [s]$ there is a subset $S_a \subseteq [t]$ such that $I_a = \cup_{b \in S_a} J_b$.

For $\mathcal{I} = \{I_i\}_{i=1}^r$ and $\mathcal{I}' = \{I'_i\}_{i=1}^s$ two partitions of $[n]$ into r and s intervals respectively, we say that the *common refinement* of $\mathcal{I}$ and $\mathcal{I}'$ is the partition $\mathcal{J}$ of $[n]$ into intervals obtained from $\mathcal{I}$ and $\mathcal{I}'$ in the obvious way, by taking all possible nonempty intervals of the form $I_i \cap I'_j$. It is clear that $\mathcal{J}$ is both a refinement of $\mathcal{I}$ and of $\mathcal{I}'$ and that $\mathcal{J}$ contains at most $r + s$ intervals.

2.1 Basic Tools.

We recall some basic tools from probability.

The VC inequality. Given a family of subsets $\mathcal{A}$ over $[n]$, define $\|p\|_{\mathcal{A}} = \sup_{A \in \mathcal{A}} |p(A)|$. The *VC dimension* of $\mathcal{A}$ is the maximum size of a subset $X \subset [n]$ that is shattered by $\mathcal{A}$ (a set X is shattered by $\mathcal{A}$ if for every $Y \subseteq X$, some $A \in \mathcal{A}$ satisfies $A \cap X = Y$).

Theorem 2.1 (VC inequality, [DL01, p.31])**.** *Let $\widehat{p}_m$ be an empirical distribution of m samples from p. Let $\mathcal{A}$ be a family of subsets of VC dimension d. Then $\mathbb{E}[\|p - \widehat{p}_m\|_{\mathcal{A}}] \leq O(\sqrt{d/m})$.*

Uniform convergence. We will also use the following uniform convergence bound:

Theorem 2.2 ([DL01, p17])**.** *Let $\mathcal{A}$ be a family of subsets over $[n]$, and $\widehat{p}_m$ be an empirical distribution of m samples from p. Let X be the random variable $\|p - \hat{p}\|_{\mathcal{A}}$. Then we have $\Pr[X - \mathbb{E}[X] > \eta] \leq e^{-2m\eta^2}$.*

3 Learning mixtures of (ε, t)-flat distributions

In this section we present and analyze our general algorithm for learning mixtures of (ε, t)-flat distributions. We proceed in stages by considering three increasingly demanding learning scenarios, each of which builds on the previous one.

3.1 First scenario: known flat decomposition.

We start with the simplest scenario, in which the learning algorithm is given a partition $\mathcal{P}$ which is a (p, ε, t)-flat decomposition of $[n]$ for the target distribution p being learned.

> Algorithm LEARN-KNOWN-DECOMPOSITION$(p, \mathcal{P}, \varepsilon, \delta)$:
> **Input:** sample access to unknown distribution p over $[n]$; (p, ε, t)-flat decomposition $\mathcal{P}$ of $[n]$; accuracy parameter ε; confidence parameter δ
>
> 1. Draw $m = O((t + \log 1/\delta)/\varepsilon^2)$ samples to obtain an empirical distribution $\widehat{p}_m$.
> 2. Return $(\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}$.

Theorem 3.1. *Let p be any unknown target distribution over $[n]$ and $\mathcal{P}$ be any (p, ε, t)-flat decomposition of $[n]$. Algorithm* LEARN-KNOWN-DECOMPOSITION$(p, \mathcal{P}, \varepsilon, \delta)$ *draws $O((t + \log(1/\delta))/\varepsilon^2)$ samples from p and with probability at least $1 - \delta$, outputs $(\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}$ such that $d_{\mathrm{TV}}((\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}, p) \leq 2\varepsilon$. Its running time is $\tilde{O}((t + \log(1/\delta)) \cdot \log(n)/\varepsilon^2)$ bit operations.*

Proof. We have $d_{\mathrm{TV}}(p, (\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}) \leq d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) + d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{P})}, (\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})})$. The first term on the right-hand side is at most ϵ by the definition of a (p, ε, t)-flat decomposition. The second term is also at most ϵ, as follows by Proposition 3.1, stated and proved below. □

Proposition 3.1. *Let p be any distribution over $[n]$ and let $\widehat{p}_m$ be an empirical distribution of $m = \Theta((s + \log 1/\delta)/\varepsilon^2)$ samples from p. Let $\mathcal{P}$ be any partition of $[n]$ into at most s intervals. Then with probability at least $1 - \delta$, $d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{P})}, (\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}) \leq \varepsilon$.*

Proof. By definition we have $d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{P})}, (\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}) = (1/2)\sum_{I\in\mathcal{P}} |p(I) - \widehat{p}_m(I)| = |p(A) - \widehat{p}_m(A)|$, where $A = \bigcup\{I \in \mathcal{P} \mid p(I) > \widehat{p}_m(I)\}$. Since $\mathcal{P}$ contains at most s intervals, A is a union of at most s intervals. Consequently the above right-hand side is at most $\|p - \widehat{p}_m\|_{\mathcal{A}_s}$, where $\mathcal{A}_s$ is the family of all unions of at most s intervals over $[n]$.[1] Since the VC-dimension of $\mathcal{A}_s$ is $2s$, Theorem 2.1 implies that the considered quantity has expected value at most ε. The claimed result now follows by applying Theorem 2.2 with $\eta = \varepsilon$. □

3.2 Second scenario: unknown flat distribution. The second algorithm deals with the scenario in which the target distribution p is $(\varepsilon/4, t)$-flat but no flat decomposition is provided to the learner. We show that in such a setting we can construct a $(p, \varepsilon, O(t/\varepsilon))$-flat decomposition $\mathcal{P}$ of $[n]$, and then we can simply use this $\mathcal{P}$ to run LEARN-KNOWN-DECOMPOSITION.

The basic subroutine RIGHT-INTERVAL will be useful here (and later). It takes as input an explicit description of a distribution q over $[n]$, an interval $J = [a, b] \subseteq [n]$, and a threshold $\tau > 0$. It returns the longest interval in $[a, b]$ that ends at b and has mass at most τ under q. If no such interval exists then $q(b)$ must exceeds τ, and the subroutine simply returns the singleton interval $[b, b]$.

> Subroutine RIGHT-INTERVAL(q, J, τ):
> **Input:** explicit description of distribution q; interval $J = [a, b]$; threshold τ
> If $q(b) > \tau$ then set $i' = b$, otherwise set $i' = \min\{a \leq i \leq b \mid q([i, b]) \leq \tau\}$. Return $[i', b]$.

The algorithm to construct a decomposition is given below:

> Algorithm CONSTRUCT-DECOMPOSITION$(p, \tau, \varepsilon, \delta)$:
> **Input:** sample access to unknown distribution p over $[n]$; parameter τ; accuracy parameter ε; confidence parameter δ
>
> 1. Draw $m = O((1/\tau + \log 1/\delta)/\varepsilon^2)$ samples to obtain an empirical distribution $\widehat{p}_m$.
> 2. Set $J = [n], \mathcal{P} = \emptyset$.
> 3. While $J \neq \emptyset$:
>   (a) Let I be the interval returned by RIGHT-INTERVAL$(\widehat{p}_m, J, \tau)$.
>   (b) Add I to $\mathcal{P}$ and set $J = J \setminus I$.
> 4. Return $\mathcal{P}$.

Theorem 3.2. *Let $\mathfrak{C}$ be a class of $(\varepsilon/4, t)$-flat distributions over $[n]$. Then for any $p \in \mathfrak{C}$, Algorithm* CONSTRUCT-DECOMPOSITION$(p, \varepsilon/(4t), \varepsilon, \delta)$ *draws $O(t/\epsilon^3 + \log(1/\delta)/\varepsilon^2)$ samples from p, and with probability at least $1 - \delta$ outputs a $(p, \varepsilon, 8t/\varepsilon)$-flat decomposition $\mathcal{P}$ of $[n]$. Its running time is $\tilde{O}((1/\tau + \log(1/\delta)) \cdot \log(n)/\varepsilon^2)$ bit operations.*

[1] Formally, define $\mathcal{A}_1 = \{[a, b] \mid 1 \leq a \leq b \leq n\} \cup \{\emptyset\}$ as the collection of all intervals over $[n]$, including the empty interval. Then $\mathcal{A}_s = \{I_1 \cup \cdots \cup I_s \mid I_1, \ldots, I_s \in \mathcal{A}_1\}$.

To prove the above theorem we will need the following elementary fact about refinements:

Lemma 3.1 ([DDS+11, Lemma 4])**.** *Let p be any distribution over $[n]$ and let $\mathcal{I} = \{I_i\}_{i=1}^{t}$ be a (p, ϵ, t)-flat decomposition of $[n]$. If $\mathcal{J} = \{J_i\}_{i=1}^{t'}$ is a refinement of $\mathcal{I}$, then $\mathcal{J}$ is a $(p, 2\epsilon, t')$-flat decomposition of $[n]$.*

We will also use the following simple observation about the RIGHT-INTERVAL subroutine:

Observation 3.1. *Suppose* RIGHT-INTERVAL(q, J, τ) *returns an interval $I \neq J$ and* RIGHT-INTERVAL$(q, J \setminus I, \tau)$ *returns I'. Then $q(I) + q(I') > \tau$.*

Proof of Theorem 3.2. Let $\tau \stackrel{\text{def}}{=} \varepsilon/(4t)$. By Observation 3.1, the partition $\mathcal{P}$ that the algorithm constructs must contain at most $2/\tau$ intervals. Let $\mathcal{Q}$ be the common refinement of $\mathcal{P}$ and a $(p, \epsilon/4, t)$-flat decomposition of $[n]$ (the existence of such a decomposition is guaranteed because every distribution in $\mathfrak{C}$ is $(\varepsilon/4, t)$-flat). Now note that $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) \leq d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{Q})}) + d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{Q})}, p^{\mathrm{flat}(\mathcal{P})})$. Since $\mathcal{Q}$ is a refinement of the $(p, \epsilon/4, t)$-flat decomposition of $[n]$, Lemma 3.1 implies that the first term on the RHS is at most $\varepsilon/2$. It remains to bound $\Delta = d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{Q})}, p^{\mathrm{flat}(\mathcal{P})})$. Fix any interval $I \in \mathcal{P}$ and let us consider the contribution $(1/2)\sum_{j \in I} |p^{\mathrm{flat}(\mathcal{Q})}(j) - p^{\mathrm{flat}(\mathcal{P})}(j)|$ of I to Δ. If $I \in \mathcal{P} \cap \mathcal{Q}$ then the contribution to Δ is zero; on the other hand, if $I \in \mathcal{P} \setminus \mathcal{Q}$ then the contribution to Δ is at most $p(I)/2$. Thus the total contribution summed across all $I \in \mathcal{P}$ is at most $(1/2)\sum_{I \in \mathcal{P} \setminus \mathcal{Q}} p(I)$. Now we observe that with probability at least $1 - \delta$ we have

$$(1/2) \sum_{I \in \mathcal{P} \setminus \mathcal{Q}} p(I) \leq \varepsilon/4 + (1/2) \sum_{I \in \mathcal{P} \setminus \mathcal{Q}} \widehat{p}_m(I), \tag{1}$$

where the inequality follows from the fact that $d_{\mathrm{TV}}(p^{\mathrm{flat}(\mathcal{P})}, (\widehat{p}_m)^{\mathrm{flat}(\mathcal{P})}) \leq \varepsilon/4$ by Proposition 3.1. If $I \in \mathcal{P} \setminus \mathcal{Q}$ then I cannot be a singleton, and hence $\widehat{p}_m(I) \leq \tau$ by definition of RIGHT-INTERVAL. Finally, it is easy to see that at most t intervals I in $\mathcal{P}$ do not belong to $\mathcal{Q}$ (because $\mathcal{Q}$ is the common refinement of $\mathcal{P}$ and a partition of $[n]$ into at most t intervals). Thus the second term on RHS of Eq. (1) is at most $t\tau = \varepsilon/4$. Hence $\Delta \leq \varepsilon/2$ and the theorem is proved. □

Our algorithm to learn an unknown $(\varepsilon/4, t)$-flat distribution is now very simple:

Algorithm LEARN-UNKNOWN-DECOMPOSITION$(p, t, \varepsilon, \delta)$:

Input: sample access to unknown distribution p over $[n]$; parameter t; accuracy parameter ε; confidence parameter δ

1. Run CONSTRUCT-DECOMPOSITION$(p, \varepsilon/(4t), \varepsilon, \delta/2)$ to obtain a $(p, \varepsilon, 8t/\varepsilon)$-flat decomposition $\mathcal{P}$ of $[n]$.
2. Run LEARN-KNOWN-DECOMPOSITION$(p, \mathcal{P}, \varepsilon, \delta/2)$ and return the hypothesis h that it outputs.

The following is now immediate:

Theorem 3.3. *Let $\mathfrak{C}$ be a class of $(\varepsilon/4, t)$-flat distributions over $[n]$. Then for any $p \in \mathfrak{C}$, Algorithm* LEARN-UNKNOWN-DECOMPOSITION$(p, t, \varepsilon, \delta)$ *draws $O(t/\epsilon^3 + \log(1/\delta)/\varepsilon^2)$ samples from p, and with probability at least $1 - \delta$ outputs a hypothesis distribution h satisfying $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}(\log(n) \cdot (t/\varepsilon^3 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

3.3 Main result (third scenario): learning a mixture of flat distributions. We have arrived at the scenario of real interest to us, namely learning an unknown mixture of k distributions each of which has an (unknown) flat decomposition. The key to learning such distributions is the following structural result, which says that any such mixture must itself have a flat decomposition:

Lemma 3.2. *Let $\mathfrak{C}$ be a class of (ε, t)-flat distributions over $[n]$, and let p be any k-mixture of distributions in $\mathfrak{C}$. Then p is a $(2\varepsilon, kt)$-flat distribution.*

Proof. Let $p = \sum_{j=1}^{k} \mu_j p_j$ be a k-mixture of components $p_1, \ldots, p_k \in \mathfrak{C}$. Let $\mathcal{P}_j$ denote the (p_j, ϵ, t)-flat decomposition of $[n]$ corresponding to p_j, and let $\mathcal{P}$ be the common refinement of $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_k$. It is clear that $\mathcal{P}$ contains at most kt intervals. By Lemma 3.1, $\mathcal{P}$ is a $(p_j, 2\epsilon, kt)$-flat decomposition for every p_j. Hence we have

$$d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) = d_{\mathrm{TV}}(\sum_{j=1}^{k} \mu_j p_j, \sum_{j=1}^{k} \mu_j (p_j)^{\mathrm{flat}(\mathcal{P})}) \leq \sum_{j=1}^{k} \mu_j d_{\mathrm{TV}}(p_j, (p_j)^{\mathrm{flat}(\mathcal{P})}) \leq 2\varepsilon. \qquad \square$$

Given Lemma 3.2, the desired mixture learning algorithm follows immediately from the results of the previous subsection:

Corollary 3.1. *Let $\mathfrak{C}$ be a class of $(\varepsilon/8, t)$-flat distributions over $[n]$, and let p be any k-mixture of distributions in $\mathfrak{C}$. Then Algorithm* LEARN-UNKNOWN-DECOMPOSITION$(p, kt, \varepsilon, \delta)$ *draws $O(kt/\varepsilon^3 + \log(1/\delta)/\varepsilon^2)$ samples from p, and with probability at least $1-\delta$ outputs a hypothesis distribution h satisfying $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}(\log(n) \cdot (kt/\varepsilon^3 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

4 Learning mixtures of log-concave distributions

In this section we apply our general method from Section 3 to learn *log-concave* distributions over $[n]$ and mixtures of such distributions.

Definition 4.1. *A probability distribution p over $[n]$ is said to be* log-concave *if it satisfies the following conditions: (i) if $1 \leq i < j < k \leq n$ are such that $p(i)p(k) > 0$ then $p(j) > 0$; and (ii) $p(k)^2 \geq p(k-1)p(k+1)$ for all $k \in [n]$.*

We note that while some of the literature on discrete log-concave distributions states that the definition consists solely of item (ii) above, item (i) is in fact necessary as well since without it log-concave distributions need not even be unimodal (see the discussion following Definition 2.3 of [FBR11]).

In Section 4.1 we give an efficient algorithm which constructs an $(\varepsilon, O(\log(1/\varepsilon)/\varepsilon))$-flat decomposition of any target log-concave distribution. Combining this with Algorithm LEARN-KNOWN-DECOMPOSITION we obtain an $\tilde{O}(1/\varepsilon^3)$-sample algorithm for learning a single discrete log-concave distribution, and combining it with Corollary 3.1 we obtain a $k \cdot \tilde{O}(1/\varepsilon^4)$-sample algorithm for learning a k-mixture of log-concave distributions.

4.1 Constructing a flat decomposition given samples from a log-concave distribution. We recall the well-known fact that log-concavity implies unimodality (see e.g. [KG71]). Thus, it is useful to analyze log-concave distributions which additionally are monotone (since a general log-concave distribution can be viewed as consisting of two such pieces). With this motivation we give the following lemma (for the proof see Appendix B.1):

Lemma 4.1. *Let p be a distribution over $[n]$ that is non-decreasing and log-concave on $[1, b] \subseteq [n]$. Let $I = [a, b]$ be an interval of mass $p(I) = \tau$, and suppose that the interval $J = [1, a-1]$ has mass $p(J) = \sigma > 0$. Then*

$$p(b)/p(a) \leq 1 + \tau/\sigma.$$

We will also use the following elementary fact:

Fact 4.1. *Let p be a distribution over $[n]$ and $I \subseteq [n]$ be an interval such that $\max_{i,j \in I} p(i)/p(j) \leq 1+\eta$. Then the flattened sub-distribution $p^{\mathrm{flat}(I)}(i) \stackrel{\mathrm{def}}{=} p(I)/|I|$ satisfies $d_{\mathrm{TV}}(p^I, p^{\mathrm{flat}(I)}) \leq \eta \cdot p(I)$.*

We are now ready to present and analyze our algorithm DECOMPOSE-LOG-CONCAVE that draws samples from an unknown log-concave distribution and outputs a flat decomposition. The algorithm simply runs the general algorithm CONSTRUCT-DECOMPOSITION with an appropriate choice of parameters. However the analysis will not go via the "generic" Theorem 3.2 (which would yield a weaker bound) but instead uses Lemma 4.1, which is specific to log-concave distributions.

Algorithm DECOMPOSE-LOG-CONCAVE(p, ε, δ):

Input: sample access to unknown log-concave distribution p over $[n]$; accuracy parameter ε; confidence parameter δ

Set $\tau = \Theta(\varepsilon/\log(1/\varepsilon))$. Run CONSTRUCT-DECOMPOSITION$(p, \tau, \varepsilon, \delta)$ and return the decomposition $\mathcal{P}$ that it yields.

Our main theorem in this section is the following (see Appendix B.2 for the proof).

Theorem 4.2. *For any log-concave distribution p over $[n]$, Algorithm* DECOMPOSE-LOGCONCAVE(p, ε, δ) *draws $O(\log(1/\varepsilon)/\varepsilon^3 + \log(1/\delta)(\log(1/\varepsilon))^2/\varepsilon^2)$ samples from p and with probability at least $1-\delta$ constructs a decomposition $\mathcal{P}$ that is $(p, \varepsilon, O(\log(1/\varepsilon)/\varepsilon))$-flat.*

Our claimed upper bounds follow from the above theorem by using our framework of Section 3. In Appendix B.3 we give the details and also show the claimed lower bounds.

5 Learning mixtures of MHR distributions

In this section we apply our general method from Section 3 to learn *monotone hazard rate (MHR)* distributions over $[n]$ and mixtures of such distributions.

Definition 5.1. *Let p be a distribution supported in $[n]$. The* hazard rate *of p is the function $H(i) \stackrel{\mathrm{def}}{=} \frac{p(i)}{\sum_{j \geq i} p(j)}$; if $\sum_{j \geq i} p(j) = 0$ then we say $H(i) = +\infty$. We say that p has* monotone hazard rate *(MHR) if $H(i)$ is a non-decreasing function over $[n]$.*

It is known that every log-concave distribution over $[n]$ is MHR but the converse is not true, as can easily be seen from the fact that every monotone non-decreasing distribution over $[n]$ is MHR.

In Section 5.1 we prove that every MHR distribution over $[n]$ has an $(\varepsilon, O(\log(n/\varepsilon)/\varepsilon))$-flat decomposition. We combine this with our general results from Section 3 to get learning results for mixtures of MHR distributions.

5.1 Learning a single MHR distribution. Our algorithm to construct a flat decomposition of an MHR distribution p is DECOMPOSE-MHR, given below. Note that this algorithm takes an explicit description of p as input and does not draw any samples from p. Roughly speaking, the algorithm works by partitioning $[n]$ into intervals such that within each interval the value of p never deviates from its value at the leftmost point of the interval by a multiplicative factor of more than $(1+\varepsilon/8)$.

Algorithm DECOMPOSE-MHR(p, ε):

Input: explicit description of MHR distribution p over $[n]$; accuracy parameter $\varepsilon > 0$

1. Set $J = [n]$ and set $\mathcal{Q}$ to be the empty set.
2. Let I be the interval returned by RIGHT-INTERVAL$(p, J, \varepsilon/8)$, and I' be the interval returned by RIGHT-INTERVAL$(p, J \setminus I, \varepsilon/8)$. Set $J = J \setminus (I \cup I')$.
3. Set $i \in J$ to be the smallest integer such that $p(i) \geq \varepsilon/(4n)$. If no such i exists, let $I'' = J$ and then go to Step 5. Otherwise, let $I'' = [1, i-1]$ and $J = J \setminus I''$.
4. While $J \neq \emptyset$:
   (a) Let $j \in J$ be the smallest integer such that either $p(j) > (1+\varepsilon/8)p(i)$ or $p(j) < \frac{1}{1+\varepsilon/8}p(i)$ holds. If no such j exists let $I''' = J$, otherwise, let $I''' = [i, j-1]$.
   (b) Add I''' to $\mathcal{Q}$, and set $J = J \setminus I'''$.
   (c) Let $i = j$.
5. Return $\mathcal{P} = \mathcal{Q} \cup \{I, I', I''\}$.

Our first lemma for the analysis of DECOMPOSE-MHR states that MHR distributions satisfy a condition that is similar to being monotone non-decreasing (see Appendix C.1 for the proof):

Lemma 5.1. *Let p be an MHR distribution over $[n]$. Let $I = [a, b] \subset [n]$ be an interval, and $R = [b+1, n]$ be the elements to the right of I. Let $\eta \stackrel{\text{def}}{=} p(I)/p(R)$. Then $p(b+1) \geq \frac{1}{1+\eta}p(a)$.*

Let $\mathcal{Q} = \{I_1, I_2, \ldots, I_{|\mathcal{Q}|}\}$, with $I_i = [a_i, b_i]$, $1 \leq i \leq |\mathcal{Q}|$, where $a_i < a_{i+1}$. Let $\mathcal{Q}' = \{I_i \in \mathcal{Q} : p(a_i) > p(a_{i+1})\}$ and $\mathcal{Q}'' = \{I_i \in \mathcal{Q} : p(a_i) \leq p(a_{i+1})\}$. Thus $\mathcal{Q}'$ consists of those intervals I in $\mathcal{Q}$ which are such that the following interval's initial value is significantly smaller than the initial value of I, and $\mathcal{Q}''$ consists of those $I \in \mathcal{Q}$ for which the following interval's initial value is significantly larger than the initial value of I. We also denote $R_i = [a_{i+1}, n]$. For convenience, we also let $a_{|\mathcal{Q}|+1} = b_{|\mathcal{Q}|} + 1$.

We first bound the "total multiplicative decrease in p" across all intervals in $\mathcal{Q}'$ (proof in Appendix C.2):

Lemma 5.2. *We have $\prod_{I_i \in \mathcal{Q}'} \frac{p(a_i)}{p(a_{i+1})} \leq \frac{8}{\epsilon}$.*

At this point we can bound the number of intervals produced by DECOMPOSE-MHR:

Lemma 5.3. *Step 4 of Algorithm* DECOMPOSE-MHR *adds at most $O(\log(n/\epsilon)/\epsilon)$ intervals to $\mathcal{Q}$.*

Proof. We first bound the number of intervals in $\mathcal{Q}'$. Let the intervals in $\mathcal{Q}'$ be $I'_1, I'_2, \ldots I'_{|\mathcal{Q}'|}$, where $I'_j = [a'_j, b'_j]$ and $a'_1 > a'_2 > \cdots > a'_{|\mathcal{Q}'|}$. Observation 3.1 implies that the total probability mass $p(I \cup I')$ is at least $\epsilon/8$. Hence, $p([b'_1 + 1, n])$ is at least $\epsilon/8$ and we have $p(R'_1) \geq \varepsilon/8$. For $j \geq 1$ it holds

$$p(R'_j) \geq (\varepsilon/8)\,(1+\varepsilon/8)^{j-1}\,. \tag{2}$$

Consequently the number of intervals in $\mathcal{Q}'$ is bounded by $O(\log(1/\epsilon)/\epsilon)$.

Now we bound the number of intervals in $\mathcal{Q}''$. We consider the value of $\prod_{I_i \in \mathcal{Q}} \frac{p(a_{i+1})}{p(a_i)}$:

$$\frac{p(a_{|\mathcal{Q}|+1})}{p(a_1)} = \prod_{I_i \in \mathcal{Q}} \frac{p(a_{i+1})}{p(a_i)} = \prod_{I_i \in \mathcal{Q}''} \frac{p(a_{i+1})}{p(a_i)} \cdot \prod_{I_i \in \mathcal{Q}'} \frac{p(a_{i+1})}{p(a_i)}$$

Since $p(a_{|\mathcal{Q}|+1}) \leq 1$ and $p(a_1) \geq \varepsilon/(4n)$, the above is at most $4n/\varepsilon$; using Lemma 5.2, we get that

$$\prod_{I_i \in \mathcal{Q}''} \frac{p(a_{i+1})}{p(a_i)} \leq (4n/\varepsilon) \cdot (8/\varepsilon) = (32n/\varepsilon^2).$$

On the other hand, for every $I_i \in \mathcal{Q}''$ we have that $\frac{p(a_{i+1})}{p(a_i)} \geq (1 + \varepsilon/8)$. Consequently there can be at most $O((1/\varepsilon)\log(n/\epsilon))$ intervals in $\mathcal{Q}''$, and the proof is complete. □

It remains only to show that $\mathcal{P}$ is actually a flat decomposition of p:

Theorem 5.1. *Algorithm* DECOMPOSE-MHR *outputs a partition $\mathcal{P}$ of $[n]$ that is $(p, \varepsilon, \log(n/\varepsilon)/\varepsilon))$-flat.*

The proof is deferred to Appendix C.3. In Appendix C.4 we give the formal statements by applying our framework and explain the lower bounds.

6 Conclusions and future work

This work introduces a simple general approach to learn mixtures of "structured" distributions over discrete domains. We illustrate the usefulness of our approach by showing it yields nearly optimal algorithms for learning mixtures of three natural and well-studied classes (log-concave, MHR and unimodal). In the process we also establish novel structural properties of these classes.

Are there any other natural distribution classes for which our general framework is applicable? We suspect so. At the technical level, the linear dependence on the parameters k and t in the sample complexity of Theorem 1.1 is optimal (up to constant factors). It would be interesting to improve the dependence on $1/\varepsilon$ from cubic down to quadratic (which would be best possible) with an efficient algorithm.

References

[ACS10] Michal Adamaszek, Artur Czumaj, and Christian Sohler. Testing monotone continuous distributions on high-dimensional real cubes. In *SODA*, pages 56–65, 2010.

[AK01] S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. In *Proceedings of the 33rd Symposium on Theory of Computing*, pages 247–257, 2001.

[AM05] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proceedings of the Eighteenth Annual Conference on Learning Theory (COLT)*, pages 458–469, 2005.

[An95] M. Y. An. Log-concave probability distributions: Theory and statistical testing. Technical Report Economics Working Paper Archive at WUSTL, Washington University at St. Louis, 1995.

[Ban08] M. Banerjee. Estimating monotone, unimodal and u-shaped failure rates using asymptotic pivots. *Statistica Sinica*, 18:467–492, 2008.

[BB05] Mark Bagnoli and Ted Bergstrom. Log-concave probability and its applications. *Economic Theory*, 26(2):pp. 445–469, 2005.

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

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

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

[BKR04] Tugkan Batu, Ravi Kumar, and Ronitt Rubinfeld. Sublinear algorithms for testing monotone and unimodal distributions. In *ACM Symposium on Theory of Computing*, pages 381–390, 2004.

[BMP63] R. Barlow, A. Marshall, and F. Proschan. Properties of probability distributions with monotone hazard rate. *Annals of Mathematical Statistics*, 34(2):375–389, 1963.

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

[Das99] S. Dasgupta. Learning mixtures of gaussians. In *Proceedings of the 40th Annual Symposium on Foundations of Computer Science*, pages 634–644, 1999.

[DDS+11] C. Daskalakis, I. Diakonikolas, R.A. Servedio, G. Valiant, and P. Valiant. Testing k-modal distributions: Optimal algorithms via reductions. Arxiv report, 2011, 2011.

[DDS12a] C. Daskalakis, I. Diakonikolas, and R.A. Servedio. Learning k-modal distributions via testing. In *SODA*, pages 1371–1385, 2012.

[DDS12b] C. Daskalakis, I. Diakonikolas, and R.A. Servedio. Learning poisson binomial distributions. pages 709–728, 2012. STOC.

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

[DR09] L. D umbgen and K. Rufibach. Maximum likelihood estimation of a log-concave density and its distribution function: Basic properties and uniform consistency. *Bernoulli*, 15(1):40–68, 2009.

[DS00] S. Dasgupta and L. Schulman. A Two-round Variant of EM for Gaussian Mixtures. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 143–151, 2000.

[FBR11] H. Jankowski F. Balabdaoui and K. Rufibach. Maximum likelihood estimation and confidence bands for a discrete log-concave distribution, 2011.

[FM99] Y. Freund and Y. Mansour. Estimating a mixture of two product distributions. In *Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 183–192, 1999.

[FOS05] J. Feldman, R. O'Donnell, and R. Servedio. Learning mixtures of product distributions over discrete domains. In *Proc. 46th Symposium on Foundations of Computer Science (FOCS)*, pages 501–510, 2005.

[Fou97] A.-L. Fougères. Estimation de densités unimodales. *Canadian Journal of Statistics*, 25:375–387, 1997.

[GJ11] Piet Groeneboom and Geurt Jongbloed. Smooth and non-smooth estimates of a monotone hazard. Technical report, 2011.

[HW93] J. Huang and J. A. Wellner. Estimation of a monotone density or monotone hazard under random censoring. Technical report, 1993.

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

[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.

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

[KSV08] Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. The spectral method for general mixture models. *SIAM J. Comput.*, 38(3):1141–1156, 2008.

[Lin95] B. Lindsay. *Mixture models: theory, geometry and applications*. Institute for Mathematical Statistics, 1995.

[MV10] Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *FOCS*, pages 93–102, 2010.

[Mye81] R.B. Myerson. Optimal auction design. *Mathematics of Operations Research*, 6:58–73, 1981.

[NS60] D. J. Newman and L. Shepp. The double dixie cup problem. *The American Mathematical Monthly*, 67(1):pp. 58–61, 1960.

[Rao69] B.L.S. Prakasa Rao. Estimation of a unimodal density. *Sankhya Ser. A*, 31:23–36, 1969.

[RS05] R. Rubinfeld and R. Servedio. Testing monotone high-dimensional distributions. In *Proc. 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 147–156, 2005.

[RW84] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–202, 1984.

[Sco92] D.W. Scott. *Multivariate Density Estimation: Theory, Practice and Visualization*. Wiley, New York, 1992.

[Sil86] B. W. Silverman. *Density Estimation*. Chapman and Hall, London, 1986.

[Sta89] Richard P. Stanley. Log-concave and unimodal sequences in algebra, combinatorics, and geometry. *Annals of the New York Academy of Sciences*, 576(1):500–535, 1989.

[TSM85] D.M. Titterington, A.F.M. Smith, and U.E. Makov. *Statistical analysis of finite mixture distributions*. Wiley & Sons, 1985.

[VV11a] 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.

[VV11b] Gregory Valiant and Paul Valiant. The power of linear estimators. In *FOCS*, 2011.

[VW02] S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. In *Proceedings of the 43rd Annual Symposium on Foundations of Computer Science*, pages 113–122, 2002.

[Wal09] G. Walther. Inference and modeling with log-concave distributions. *Statistical Science*, 24(3):319–327, 2009.

[Wan86] Jane-Ling Wang. Asymptotically minimax estimators for distributions with increasing failure rate. *The Annals of Statistics*, 14(3):pp. 1113–1131, 1986.

[Weg70] E.J. Wegman. Maximum likelihood estimation of a unimodal density. I. and II. *Ann. Math. Statist.*, 41:457–471, 2169–2174, 1970.

A Proof of Proposition 1.1

At a high level, the algorithm A' works by drawing a large set of samples from the target mixture and trying all possible ways of partitioning the sample into k disjoint subsamples. For each partition of the sample it runs algorithm A over each subsample and combines the resulting hypothesis distributions (guessing the mixture weights) to obtain a hypothesis mixture distribution. Finally, a "hypothesis testing" procedure is used to identify a high-accuracy hypothesis from the collection of all hypotheses distributions obtained in this way.

More precisely, let p denote the unknown target k-mixture of distributions from $\mathfrak{C}$. Algorithm A' works as follows:

1. Draw a sample S of $M = \tilde{O}(k/\varepsilon) \cdot m(n, \varepsilon/20) \cdot \log(5k/\delta)$ samples from p.
2. For each possible way of partitioning S into k disjoint subsamples $\bar{S} = (S_1, \ldots, S_k)$ such that each $|S_i| \geq m(n, \varepsilon/20) \cdot \log(5k/\delta)$, run algorithm A a total of k times, using S_i as the input sample for the i-th run, to obtain hypothesis distributions $h_1^{\bar{S}}, \ldots, h_k^{\bar{S}}$. For each vector $\mu = (\mu_1, \ldots, \mu_k)$ of non-negative mixing weights that sum to 1 and satisfy $\mu_i = (\text{integer}) \cdot \varepsilon/(20k)$, let $h_\mu^{\bar{S}}$ be the mixture distribution $\sum_{i=1}^k \mu_i h_i^{\bar{S}}$.
3. Draw $M' = O(M \log k + k \log(k/\varepsilon)) \cdot \log(5/\delta)/\varepsilon^2$ samples from p and use them to run the "hypothesis testing" routine described in Lemma 11 of [DDS12b] over all hypotheses $h_\mu^{\bar{S}}$ obtained in the previous step. Output the hypothesis distribution that this routine outputs.

We now proceed with the analysis of the algorithm. Let $p = \sum_{i=1}^k \kappa_i p_i$ denote the target k-mixture, where $\kappa_1, \ldots, \kappa_k$ are the mixing weights and $p_1, \ldots, p_k$ are the components. Without loss of generality we may assume that $i = 1, \ldots, \ell$ are the components such that the mixing weights $\kappa_1, \ldots, \kappa_\ell$ are at least $\varepsilon/(20k)$. A standard "balls in bins" analysis (see [NS60]) implies that with probability at least $1 - \delta/5$ the sample S contains at least $m(n, \varepsilon/20) \cdot \log(5k/\delta)$ draws from each component $p_1, \ldots, p_\ell$; we assume going forth that this is indeed the case. Thus there will be some partition $\bar{S} = (S_1, \ldots, S_k)$ which is such that each S_i with $1 \leq i \leq \ell$ consists entirely of samples drawn from the component p_i. For this $\bar{S}$, we have that with failure probability at most $(\delta/(5k)) \cdot k \leq \delta/5$, each hypothesis distribution $h_i^{\bar{S}}$ for $1 \leq i \leq \ell$ satisfies $d_{\mathrm{TV}}(p_i, h_i^{\bar{S}}) \leq \varepsilon/20$. Now let $\mu^\star$ denote the vector of hypothesis mixing weights (as described in Step 2) that has $|\mu_i^\star - \kappa_i| \leq \varepsilon/(20k)$ for all $i = 1, \ldots, k$. It is not difficult to show that the hypothesis mixture distribution $h^\star = h_{\mu^\star}^{\bar{S}}$ satisfies $d_{\mathrm{TV}}(h^\star, p) \leq 3\varepsilon/20 < \varepsilon/6$, where $\varepsilon/20$ comes from the errors $d_{\mathrm{TV}}(p_i, h_i^{\bar{S}})$ for $i \leq \ell$, $\varepsilon/20$ comes from the inaccuracy in the mixing weights, and $\varepsilon/20$ comes from the (at most k) components p_j with $j > \ell$ that each have mixing weight at most $\varepsilon/(20k)$.

Thus we have established that there is at least one hypothesis distribution $h^\star$ among the $h_\mu^{\bar{S}}$'s that has $d_{\mathrm{TV}}(p, h^\star) \leq \varepsilon/6$. There are at most $N = k^M \cdot (20k/\varepsilon)^k$ hypotheses $h_\mu^{\bar{S}}$ generated in Step 2, so the algorithm of Lemma 11 of [DDS12b] requires $O(\log N) \log(5/\delta)/\varepsilon^2 \leq M'$ samples, and with probability at least $1 - \delta/5$ it outputs a hypothesis distribution h that has $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. The overall probability of outputting an ε-accurate hypothesis is at least $1 - \delta$, and the proposition is proved.

B Omitted Details from Section 4

B.1 Proof of Lemma 4.1. Let $s \stackrel{\text{def}}{=} |I| = b - a + 1$ be the length of I. We decompose J into intervals $J_1, \ldots, J_t$ of length s, starting from the right. More precisely, $J_j \stackrel{\text{def}}{=} I - js = [a - js, b - js]$ for $1 \le j \le t \stackrel{\text{def}}{=} \lceil (a-1)/s \rceil$. The leftmost interval J_t may contain non-positive integers; for this reason define $p(i) \stackrel{\text{def}}{=} 0$ for non-positive i (the new distribution is still log-concave). Also define $J_0 \stackrel{\text{def}}{=} I = [a, b]$. Let $\lambda \stackrel{\text{def}}{=} p(b)/p(a)$. We claim that

$$p(i - s) \le (1/\lambda) \cdot p(i) \tag{3}$$

for $1 \le i \le b$. Eq. (3) holds for $i = b$, since $p(b - s) \le p(a)$ by the non-decreasing property. The general case $i \le b$ follows by induction and using the fact that the ratio $p(i - 1)/p(i)$ is non-decreasing in i for any log-concave distribution (an immediate consequence of the definition of log-concavity).

It is easy to see that Eq. (3) implies

$$p(J_{j+1}) \le (1/\lambda) \cdot p(J_j)$$

for $0 \le j \le t$. Since the intervals have geometrically decreasing mass, this implies that

$$\sigma = \sum_{1 \le j \le t} p(J_j) \le p(I) \sum_{j \ge 1} \lambda^{-j} = \frac{\tau}{\lambda - 1}.$$

Rearranging yields the desired inequality.

B.2 Proof of Theorem 4.2. We first note that the number of intervals in $\mathcal{P}$ is at most $2/\tau$ by Observation 3.1; this will be useful below. We may also assume that $d_{\mathrm{K}}(p, \widehat{p}_m) \le \tau$, where $\widehat{p}_m$ is the empirical distribution obtained in Step 1 of CONSTRUCT-DECOMPOSITION; this inequality holds with probability at least $1 - \delta$, as follows by a combined application of Theorems 2.1 and 2.2. Since p is log-concave, it is unimodal. Let i_0 be a mode of p.

Let $\mathcal{P}_L = \{I \in \mathcal{P} \mid I \subset [1, i_0 - 1]\}$ be the collection of intervals to the left of i_0. We now bound the contribution of intervals in $\mathcal{P}_L$ to $\Delta \stackrel{\text{def}}{=} d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})})$. Let $I_1, \ldots, I_{t_L}$ be the intervals in $\mathcal{P}_L$ listed from left to right. Let $J_j = \cup_{j' < j} I_{j'}$ be the union of intervals to the left of I_j. If I_j is a singleton, its contribution to Δ is zero. Otherwise,

$$p(I_j) \le \widehat{p}_m(I_j) + \tau \le 2\tau$$

by the τ-closeness of p and $\widehat{p}_m$ in Kolmogorov distance and the definition of RIGHT-INTERVAL. Also, by Observation 3.1, $\widehat{p}_m(J_j) \ge \lfloor (j-1)/2 \rfloor \tau \ge ((j-1)/2 - 1)\tau$, and hence

$$p(J_j) \ge \widehat{p}_m(J_j) - \tau \ge \tau(j - 5)/2,$$

again by closeness in Kolmogorov distance.

Since p is non-decreasing on $[1, i_0 - 1]$, we have

$$\|p^{I_j} - p^{\mathrm{flat}(I_j)}\|_1 \le \frac{8}{j-5}\tau$$

for $j > 5$, by Lemma 4.1 and Fact 4.1, using the upper and lower bounds on $p(I_j)$ and $p(J_j)$ respectively. Consequently, $\|p^{I_j} - p^{\mathrm{flat}(I_j)}\|_1 \le O(\tau/j)$ for all $j \in [t_L]$. Summing this inequality, we get

$$\sum_{j \le t_L} \|p^I - p^{\mathrm{flat}(I_j)}\|_1 \le \sum_{j \le t_L} O(\tau/j) = O(\tau \log(1/\tau)).$$

The right-hand side is at most $\varepsilon/2$ by our choice of τ (with an appropriate constant in big-theta).

Similarly, let $\mathcal{P}_R = \{I \in \mathcal{P} \mid I \subset [i_0+1, n]\}$ be the collection of intervals to the right of i_0. An identical analysis (using the obvious analogue of Lemma 4.1 for non-increasing log-concave distributions on $[i_0+1, n]$) shows that the contribution of intervals in $\mathcal{P}_R$ to Δ is at most $\varepsilon/2$.

Finally, let $I_0 \in \mathcal{P}$ be the interval containing i_0. If I_0 is a singleton, it does not contribute to Δ. Otherwise, $\widehat{p}_m(I_0) \leq \tau$ and $p(I_0) \leq 2\tau$, hence the contribution of I_0 to Δ is at most 2τ.

Combining all three cases, $\|p - p^{\mathrm{flat}(\mathcal{P})}\|_1 \leq \varepsilon/2 + \varepsilon/2 + 2\tau \leq 2\varepsilon$. Hence $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) \leq \varepsilon$ as was to be shown.

B.3 Formal Theorem Statements. It is clear that we can learn any unknown log-concave distribution by running Algorithm DECOMPOSE-LOG-CONCAVE$(p, \varepsilon, \delta/2)$ to obtain a decomposition $\mathcal{P}$ and then Algorithm LEARN-KNOWN-DECOMPOSITION$(p, P, \varepsilon, \delta/2)$ to obtain a hypothesis distribution h:

Corollary B.1. *Given sample access to a log-concave distribution p over $[n]$, there is an algorithm* LEARN-LOG-CONCAVE(p, ε, δ) *that uses $O(\log(1/\delta)\log(1/\varepsilon)/\varepsilon^3)$ samples from p and with probability at least $1-\delta$ outputs a distribution h such that $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}(\log(n) \cdot (1/\varepsilon^3 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

Theorem 4.2 of course implies that every log-concave distribution p is $(\varepsilon, O(\log(1/\varepsilon)/\varepsilon))$-flat. We may thus apply Corollary 3.1 and obtain our main learning result for k-mixtures of log-concave distributions:

Corollary B.2 (see Theorem 1.2)**.** *Let p be any k-mixture of log-concave distributions over $[n]$. There is an algorithm* LEARN-LOG-CONCAVE-MIXTURE$(p, k, \varepsilon, \delta)$ *that draws $O(k\log(1/\varepsilon)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2)$ samples from p and with probability at least $1-\delta$ outputs a distribution h such that $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}(\log(n) \cdot (k\log(1/\varepsilon)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

Lower bounds. It is shown in [DL01, Lemma 15.1] that learning a continuous distribution whose density is bounded and convex over $[0,1]$ to accuracy ε requires $\Omega((1/\varepsilon)^{5/2})$ samples. An easy adaptation of this argument implies the same result for a bounded concave density over $[0,1]$. By an appropriate discretization procedure, one can show that learning a discrete concave density over $[n]$ requires $\Omega((1/\varepsilon)^{5/2})$ samples for all $\varepsilon \geq 1/n^{\Omega(1)}$. Since a discrete concave distribution is also log-concave, the same lower bound holds for this case too. For the case of k-mixtures, we may consider a uniform mixture of k component distributions where the i-th distribution in the mixture is supported on $[1+(i-1)n/k, in/k]$ and is log-concave on its support. It is clear that each component distribution is log-concave over $[n]$, and it is not difficult to see that in order to learn such a mixture to accuracy ε, at least $9/10$ of the component distributions must be learned to total variation distance at most 10ε. We thus get that for $k \leq n^{1-\Omega(1)}$ and $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a mixture of k log-concave distributions to accuracy ε must use $\Omega(k\varepsilon^{-5/2})$ samples.

C Omitted Details from Section 5

C.1 Proof of Lemma 5.1. If $p(b+1) > p(a)$ then $p(b+1) \geq \frac{1}{1+\eta}p(a)$ holds directly, so for the rest of the proof we may assume that $p(b+1) \leq p(a)$.

By the definition of the MHR condition we have $\frac{p(a)}{p([a+1,n])} \leq \frac{p(b+1)}{p([b+2,n])}$ and hence we have

$$\frac{p([a,n])}{p(a)} \geq \frac{p([b+1,n])}{p(b+1)}.$$

Thus

$$p(b+1) \geq \frac{p([b+1,n])}{p([a,n])}p(a) = \frac{1}{1+\eta}p(a)$$

as desired.

C.2 Proof of Lemma 5.2. Observation 3.1 implies that the total probability mass $p(I \cup I')$ on intervals I and I' is at least $\epsilon/8$. We thus have

$$\prod_{I_i \in \mathcal{Q}'} \frac{p(a_i)}{p(a_{i+1})} \leq \prod_{I_i \in \mathcal{Q}'} \frac{p(I_i)+p(R_i)}{p(R_i)} \leq \prod_{I_i \in \mathcal{Q}} \frac{p(I_i)+p(R_i)}{p(R_i)} = \frac{p(I_1)+p(R_1)}{p(R_{|\mathcal{Q}|})} \leq \frac{1}{\varepsilon/8},$$

where the first inequality follows from Lemma 5.1, the second inequality is self-evident, the equality follows from the telescoping product, and the final inequality is because $p(I \cup I') \geq \varepsilon/8$.

C.3 Proof of Theorem 5.1. Lemma 5.3 shows that $\mathcal{P}$ contains at most $O(\log(n/\epsilon)/\epsilon)$ intervals, so it remains only to argue that $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})}) \leq \varepsilon$.

We first consider the two rightmost intervals I and I'. If $|I| = 1$ then clearly $d_{\mathrm{TV}}(p^I, p^{\mathrm{flat}(I)}) = 0$, and if $|I| > 1$ then $p(I) \leq \frac{\epsilon}{8}$ and consequently $d_{\mathrm{TV}}(p^I, p^{\mathrm{flat}(I)}) \leq \varepsilon/8$. Identical reasoning applies to I'. For the leftmost interval I'', we have that $p(I'') \leq \frac{\epsilon}{4}$, so $d_{\mathrm{TV}}(p^{I''}, p^{\mathrm{flat}(I'')}) \leq \varepsilon/4$. Thus, so far we have shown that the contribution to $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})})$ from $I \cup I' \cup I''$ is at most $\varepsilon/2$.

Now for each interval I''' in $\mathcal{Q}$, we have

$$\max_{i,j \in I'''} \frac{p(i)}{p(j)} \leq (1+\varepsilon/8)^2 = 1 + \varepsilon/4 + \varepsilon^2/64.$$

Since the total probability mass on intervals I and I' is at least $\epsilon/8$ by Observation 3.1, the total probability mass on intervals in $\mathcal{Q}$ is at most $1 - \varepsilon/8$. An easy calculation using Observation 4.1 shows that the total contribution to $d_{\mathrm{TV}}(p, p^{\mathrm{flat}(\mathcal{P})})$ from intervals in $\mathcal{Q}$ is at most $\varepsilon/4$, and the theorem is proved.

C.4 Formal Theorem Statements. Applying Corollary 3.1, we get our main learning result for mixtures of MHR distributions:

Corollary C.1 (see Theorem 1.3)**.** *Let p be any k-mixture of MHR distributions over $[n]$. There is an algorithm* LEARN-MHR-MIXTURE$(p, k, \varepsilon, \delta)$ *that draws $O(k \log(n/\varepsilon)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2)$ samples from p and with probability at least $1 - \delta$ outputs a distribution h such that $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}((\log n)^2 \cdot (k \log(1/\varepsilon)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

Lower bounds. Birgé [Bir87a] showed that for $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a monotone distribution over $[n]$ to accuracy ε must use $\Omega(\log(n)/\varepsilon^3)$ samples. We may consider a uniform mixture of k component distributions where the i-th distribution in the mixture is supported on and monotone non-decreasing over $[1 + (i-1)n/k, in/k]$. Each component distribution is MHR, and it is not difficult to see that in order to learn such a mixture to accuracy ε at least $9/10$ of the component distributions must be learned to total variation distance at most 10ε. We thus get that for $k \leq n^{1-\Omega(1)}$ and $\varepsilon \geq 1/n^{\Omega(1)}$, any algorithm for learning a mixture of k MHR distributions to accuracy ε must use $\Omega(k \log(n)/\varepsilon^3)$ samples.

D Learning mixtures of unimodal and t-modal distributions

In this section we apply our general method from Section 3 to learn mixtures of unimodal (and, more generally, t-modal) distributions over $[n]$. Here our task is quite easy because of a result of L. Birgé [Bir87b] which essentially provides us with the desired flat decompositions. [2]

We begin by defining unimodal and t-modal distributions over $[n]$:

[2] We note that Birgé's structural result was obtained as part of an efficient learning algorithm for monotone distributions; Birgé subsequently gave an efficient learning algorithm for unimodal distributions [Bir97]. However, we are not aware of work prior to ours on learning *mixtures* of unimodal or t-modal distributions.

Definition D.1. *A distribution p over $[n]$ is* unimodal *if there exists $i \in [n]$ such that p is non-decreasing over $[1, i]$ and non-increasing over $[i, n]$. For $t > 1$, distribution p over $[n]$ is t-*modal *if there is a partition of $[n]$ into t intervals $I_1, \ldots, I_t$ such that the conditional distributions $p_{I_1}, \ldots, p_{I_t}$ are each unimodal.*

By adapting a construction of Birgé (proved in [Bir87b] for distributions over the continuous real line) to the discrete domain $[n]$, Daskalakis et al. [DDS$^+$11] established the following:

Theorem D.1 ([DDS$^+$11, Theorem 5])**.** *Let p be any monotone distribution (either non-increasing or non-decreasing) over $[n]$. Then p is $(\varepsilon, O(\log(n)/\varepsilon))$-flat.*

We note that it can be shown (using the same construction that is used in the $\Omega(\log(n)/\varepsilon^3)$ sample complexity lower bound of [Bir87a] for learning monotone distributions) that $O(\log(n)/\varepsilon)$ is the best possible bound for the number of intervals required in Theoorem D.1.

An immediate consequence of Theorem D.1 is that any unimodal distribution over $[n]$ is $(\varepsilon, O(\log(n)/\varepsilon))$-flat, and any t-modal distribution over $[n]$ is $(\varepsilon, O(t \cdot \log(n)/\varepsilon))$-flat. Using Corollary 3.1 we thus obtain the following results for learning mixtures of unimodal or t-modal distributions:

Corollary D.1 (see Theorem 1.4)**.** *For any $t \geq 1$, let p be any k-mixture of t-modal distributions over $[n]$. There is an algorithm* LEARN-MULTI-MODAL-MIXTURE$(p, k, t, \varepsilon, \delta)$ *that draws $O(kt \log(n)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2)$ samples from p and with probability at least $1-\delta$ outputs a distribution h such that $d_{\mathrm{TV}}(p, h) \leq \varepsilon$. Its running time is $\tilde{O}(\log(n) \cdot (kt \log(n)/\varepsilon^4 + \log(1/\delta)/\varepsilon^2))$ bit operations.*

Lower bounds. The lower bound arguments we gave for mixtures of MHR distributions (which are based on Birgé's lower bounds for learning monotone distributions) apply unchanged for mixtures of unimodal distributions, since every distribution which is supported on and monotone non-decreasing over $[1 + (i-1)n/k, in/k]$ is unimodal over $[n]$.