Learning mixtures of product distributions

Jon Feldman*
Industrial Engineering and Operations Research
Columbia University
jonfeld@ieor.columbia.edu

Ryan O'Donnell†
School of Mathematics
Institute for Advanced Study
odonnell@ias.edu

Rocco A. Servedio
Department of Computer Science
Columbia University
rocco@cs.columbia.edu

Abstract

In this paper we give:

- A poly(n) time algorithm for learning a mixture of any constant number of product distributions over the n-dimensional Boolean cube $\{0,1\}^n$. Previous polynomial time algorithms could only learn a mixture of two product distributions over $\{0,1\}^n$. We also give evidence that no algorithm can learn a mixture of a superconstant number of product distributions over $\{0,1\}^n$ in poly(n) time.
- A poly(n) time algorithm for learning a mixture of any constant number of axis-aligned Gaussians in $\mathbf{R}^n$ (the Gaussians need not be spherical). Our algorithm constructs a highly accurate approximation to the unknown mixture of Gaussians and, unlike previous algorithms, makes *no* assumptions about the minimum separation between the centers of the Gaussians.

We obtain both results via a new poly(n) time algorithm which, given samples from a mixture $\mathbf{Z}$ of any constant number of product distributions over $\mathbf{R}^n$, outputs a list of candidate "descriptions" at least one of which is an accurate description of $\mathbf{Z}$.

*Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship

†This material is based upon work supported in part by the National Science Foundation under agreement No. CCR-0324906. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

1 Introduction

In this paper we study *mixture distributions*. Given distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\mathbf{R}^n$ and weights $\pi^1, \ldots, \pi^k$ with $\sum \pi^i = 1$, the mixture distribution $\mathbf{Z}$ is given by first selecting i with probability π^i and then drawing a sample from $\mathbf{X}^i$. Mixture distributions arise in many practical scientific situations; indeed, as observed in the first sentence of [22], "Finite mixture distributions have been used as models throughout the history of modern statistics." As early as 1886 the Canadian astronomer S. Newcomb considered models based on mixtures of Gaussians [17], and in 1894 the mathematical biologist K. Pearson considered decomposing such mixtures by studying their moments [18].

Our work addresses the natural problem of *learning* mixtures of distributions. In this problem one is given a class $\mathcal{C}$ of distributions over $\mathbf{R}^n$ and random data sampled from an unknown mixture of k distributions from $\mathcal{C}$. The goal is to output a *hypothesis* mixture of k distributions from $\mathcal{C}$ which is very close to the unknown mixture, according to some distance measure. We will make a more precise statement of the problem in Section 2.

In this paper we learn mixtures of *product distributions* over $\mathbf{R}^n$; i.e., the classes $\mathcal{C}$ will consist of distributions $\mathbf{X}^i$ whose n coordinates are mutually independent. There are many natural examples of such distributions, and indeed two of the mixture-learning problems most frequently studied in the clustering and learning theory communities — mixtures of product distributions over $\{0,1\}^n$ and mixtures of axis-aligned n-dimensional Gaussians — fall into this category.

1.1 Our Results

The cornerstone of our learning algorithms is an efficient procedure Mix-A-Lot which takes as input samples from an unknown mixture of a constant number of product distributions and, roughly speaking, tries to output accurate estimates for all of the *mixing weights* π^i and *coordinate means* $\mu^i_j := \mathbf{E}[\mathbf{X}^i_j]$. More precisely, Mix-A-Lot outputs a *list* of poly(n/ϵ) many *candidate descriptions* $\langle \{\hat{\pi}^i\}, \{\hat{\mu}^i_j\} \rangle$, and we prove that with high probability at least one of the candidate descriptions is *parametrically accurate*; i.e. for this candidate, all $\hat{\pi}^i$'s are ϵ-close to the true π^i's and all $\hat{\mu}^i_j$'s are ϵ-close to the true μ^i_j's. (Actually, the guarantee is slightly weaker than this; see the statement of Theorem 1.)

We use algorithm Mix-A-Lot to obtain two new learning results: (i) learning mixtures of product distributions over the Boolean cube, and (ii) learning mixtures of axis-aligned Gaussians.

A product distribution $\mathbf{X}^i$ over the Boolean cube is completely specified by its coordinate means μ^i_j. As a consequence, when learning mixtures of these distributions the candidates output by Mix-A-Lot can be interpreted as true hypothesis distributions. After running Mix-A-Lot we pass the resulting hypothesis distributions through a maximum likelihood algorithm, and we show that this algorithm selects a very accurate hypothesis with high probability. Thus we give a polynomial time algorithm for learning the mixture of any constant number of product distributions over $\{0,1\}^n$.

Next we consider the case when the unknown distribution $\mathbf{Z}$ is a mixture of axis-aligned n-dimensional Gaussians. Knowing the means of Gaussians is not enough to specify them; however, knowing the means and variances is enough. By running Mix-A-Lot *twice*, once on $\mathbf{Z}$ and once on $\mathbf{Z}^2$, we get good approximations to the mixing weights, means, and second moments, from which we can get good estimates of the variances. Again we can convert these parametric descriptions into true hypothesis distributions, pass them through a maximum likelihood algorithm, and come up with a single hypothesis mixture of Gaussians which is very close to $\mathbf{Z}$. Thus we give a polynomial time algorithm for learning the mixture of any constant number of axis-aligned Gaussians in $\mathbf{R}^n$.

We note that in this case our algorithm is not strongly polynomial; the running time depends polynomially on the magnitude of the Gaussians' means and variances.

1.2 Comparison with Previous Work

There is a vast body of previous statistical work dealing with the general problem of analyzing mixture data — see [16, 20, 22] for surveys. To a large degree this work has been concerned with trying to find the best mixture model (in terms of likelihood) which explains a given data sample. Unfortunately it is well known that in the much-studied case of mixtures of Gaussians, there is no analytic solution to this problem. Further, the most popular heuristic for trying to find the best model — the EM algorithm of Dempster, Laird, and Rubin [9] — can be shown to run in exponential time in the worst case.

Recently there has been renewed algorithmic interest in the problem of learning mixtures of Gaussians from the point of view of *clustering*. In this framework, given samples drawn from a mixture of "well-separated" Gaussians, the goal is to classify each point in the sample according to which Gaussian it came from. (Note that there must be some separation requirements on the Gaussians for this goal to make sense.) Once the points have been clustered one can easily estimate the means and variances of each cluster. Dasgupta [7] gave an efficient algorithm to learn a mixture of spherical Gaussians under a strong separation requirement; this was later improved by Dasgupta and Schulman [8] who reduced the separation required between the Gaussians. Arora and Kannan [1] generalized these results to non-spherical Gaussians. Finally, Vempala and Wang [23] gave an alternate algorithm for spherical Gaussians that requires even less separation.

In our result for mixtures of Gaussians we do not make *any* minimum separation assumptions on the Gaussians. In this case one cannot hope to solve the clustering problem. Instead, we solve the natural "PAC-style" unsupervised learning problem of constructing a hypothesis mixture of Gaussians which is very close to the true distribution in KL divergence or in total variation distance. (When the Gaussians are separated, the hypothesis our algorithm constructs can easily be used to cluster.)

There are several differences in the running time of our algorithm versus earlier algorithms for mixtures of Gaussians. Our running time bound is $n^{\text{poly}(k)}$ whereas previous algorithms run in time $\text{poly}(k)$ or k^k. (As described below, there is evidence to suggest that for our definition of learning as opposed to clustering it may be very difficult to learn mixtures of $k = \omega(1)$ product distributions in $\text{poly}(n)$ time.) On the other hand, the running time of our algorithm does not depend on the magnitude of the smallest mixing weight, in contrast to several of the algorithms mentioned above.

Many researchers have also studied PAC-style learning of mixture distributions over the Boolean cube $\{0,1\}^n$. Kearns *et al.* [15] gave an efficient algorithm for learning the restricted class of mixtures of *Hamming balls*; these are product distributions in which all the coordinate means μ_j^i are either p or $1-p$ for some p fixed over all mixture components. More recently, Freund and Mansour [11] and also Cryan *et al.* [6] gave efficient algorithms for learning a mixture of two probability distributions over $\{0,1\}^n$, leaving the cases $k > 2$ as open problems. Our algorithm for solving this problem runs in time $n^{\text{poly}(k)}$ which is polynomial only if k is constant. However, we give a reduction from a notorious open question in computational learning theory to the problem of learning a mixture of any superconstant number of product distributions over $\{0,1\}^n$. Thus solving this problem for any $k = \omega(1)$ would require a major breakthrough.

1.3 Outline of This Paper

We begin in Section 2 by formally defining our learning model. In Section 3 we give a description of the algorithm Mix-A-Lot and state the theorem proving its correctness. In Section 4, we show how to convert parametric descriptions of mixtures of product distributions into true mixture distributions, in such a way that parametrically *accurate* descriptions become distributions with close KL divergence to the target distribution. Then in Section 5 we describe how a maximum-likelihood procedure can find an accurate distribution (one with good KL divergence or variation distance) among the list of converted candidate distributions, one of which is guaranteed to have good KL divergence. In Section 6 we formally state our two main results: polynomial-time algorithms for learning mixtures of any constant number of product distributions over the Boolean cube, and mixtures of any constant number of axis-aligned Gaussians. Finally, in Section 7 we give our reduction from a notorious open question in learning theory — learning $\omega(1)$-size decision trees from uniform random examples — to the problem of learning mixtures of $k = \omega(1)$ product distributions over $\{0,1\}^n$. In Section 8 we discuss the generality of our methods and directions for further research.

2 Learning Preliminaries

Our unsupervised learning framework is inspired by the Probably Approximately Correct model of learning probability distributions which was proposed by Kearns *et al.* [15]. In this framework the learning algorithm is given access to samples drawn from the target distribution $\mathbf{Z}$ to be learned, and the learning algorithm must (with high probability) output an accurate approximation $\mathbf{Z}'$ of the target distribution $\mathbf{Z}$.

To make this definition precise, we must specify some notion of the distance between two probability distributions. Following [15], we use the *Kullback-Leibler (KL) divergence* (also known as the *relative entropy*) as our distance measure. The KL divergence between $\mathbf{Z}$ and $\mathbf{Z}'$ is defined to be

$$\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') := \int_x \mathbf{Z}(x) \ln(\mathbf{Z}(x)/\mathbf{Z}'(x))$$

where here we have identified the distributions with their pdfs. (In the case of discrete distributions the integral may be viewed as a sum.) While the KL divergence is not in fact a metric, it is a stringent and commonly used measure of the distance between probability distributions. Further, it has nice properties with respect to the maximum-likelihood algorithms. It is known (see [5]) that for all distributions $\mathbf{Z}$, $\mathbf{Z}'$,

$$0 \leq \frac{\|\mathbf{Z} - \mathbf{Z}'\|_1^2}{4(\ln 2)^2} \leq \mathrm{KL}(\mathbf{Z}||\mathbf{Z}'),$$

where $\|\cdot\|_1$ denotes total variation distance, so that if the KL divergence is small, so is the variation distance.

We make the following formal definition:

Definition 1 *Let $\mathcal{D}$ be a class of probability distributions. An efficient (proper) learning algorithm for $\mathcal{D}$ is an algorithm which, given $\epsilon, \delta > 0$ and samples drawn from any distribution $\mathbf{Z} \in \mathcal{D}$, runs in poly$(n, \frac{1}{\epsilon}, \frac{1}{\delta})$ time and outputs a representation of a distribution $\mathbf{Z}' \in \mathcal{D}$ such that $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') \leq \epsilon$ with probability at least $1 - \delta$.*

Throughout the remainder of the paper we write $\mathbf{Z}$ to denote an unknown target mixture of n-dimensional product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ with mixing weights $\pi^1, \ldots, \pi^k$, where k is an arbitrary constant. We also write μ_j^i to denote $\mathbf{E}[\mathbf{X}_j^i]$.

3 The Algorithm MIX-A-LOT

In this section we give a poly(n/ϵ) time algorithm MIX-A-LOT which, given samples from $\mathbf{Z}$, outputs a list of candidates $\langle\{\hat{\pi}^1,\ldots,\hat{\pi}^k\},\{\hat{\mu}_1^1,\ldots,\hat{\mu}_n^k\}\rangle$. We show that with high probability, at least one of these candidates is an additive ϵ-accurate approximation to each of the k true mixing weights $\pi^1,\ldots,\pi^k$ and to each of the true expectations $\mu_j^i = \mathbf{E}[\mathbf{X}_j^i]$ for which the corresponding mixing weight π^i is not too small. The algorithm assumes only that the values $|\mu_j^i|$ are bounded and that $\mathbf{E}[\mathbf{Z}_j\mathbf{Z}_{j'}]$ can be estimated efficiently for each $j \neq j'$ (we make this precise below).

3.1 Tools for the algorithm: approximating and guessing

Definition 2 *Let $\mathbf{X}$ be a distribution over $\mathbf{R}$. We say that $\mathbf{X}$ is* $\lambda(\epsilon,\delta)$-samplable *if there is an algorithm $\mathcal{A}$ which, given access to draws from $\mathbf{X}$, runs for $\lambda(\epsilon,\delta)$ steps and outputs (with probability at least $1-\delta$ over the draws from $\mathbf{X}$) a quantity $\hat{\mu}$ satisfying $|\hat{\mu} - \mathbf{E}[\mathbf{X}]| \leq \epsilon$.*

Being $\lambda(\epsilon,\delta)$-samplable is a mild condition; it is easily seen from standard large deviation bounds that any distribution $\mathbf{X}$ with support bounded in $[-M,M]$ is poly$(M,\frac{1}{\epsilon},\log\frac{1}{\delta})$-samplable.

Our algorithm makes empirical estimates of the means of various $\lambda(\epsilon,\delta)$-samplable distributions. In the analysis of our algorithm we will assume that all such estimates are indeed within the desired $\pm\epsilon$ error range. This is without loss of rigor since if the algorithm makes T estimates as above, we can make all of the estimates using $\delta' = \delta/T$, and thus with probability at least $1-\delta$ all estimates will be within the desired range. This incurs a multiplicative $\log(T/\delta)$ factor toward the overall running time of the algorithm.

In describing our algorithm, we will frequently speak of *guessing* an unknown value $\alpha \in \mathbf{R}$ to within additive error $\pm\epsilon$, where α lies in the range $[-K,K]$. For simplicity we assume that $K/2\epsilon$ is an integer. When we say "Guess α to within ϵ and then do procedure P," we really mean "Do procedure P exactly $K/\epsilon+1$ times, using values $-K, -K+2\epsilon, -K+4\epsilon,\ldots,-2\epsilon,0,2\epsilon,\ldots,K-2\epsilon,K$ for α." Thus guessing α to within $\pm\epsilon$ incurs a multiplicative $(K/\epsilon+1)$ factor toward the running time of procedure P.

3.2 The idea of the algorithm

In this subsection we give an intuitive description of algorithm MIX-A-LOT to highlight the main ideas. The actual algorithm and proof of correctness are given later.

Algorithm MIX-A-LOT is given access to draws from the mixture $\mathbf{Z}$. We assume that each μ_j^i satisfies $|\mu_j^i| \leq \mu_{\max}$ and that each random variable $\mathbf{Z}_j\mathbf{Z}_{j'}$ is $\lambda(\epsilon,\delta)$-samplable for $1 \leq j < j' \leq n$.

The first step of the algorithm is to guess each mixing weight $\pi^1,\ldots,\pi^k$ to within $\pm\epsilon_{\text{wts}}$. This incurs a factor of $(1/\epsilon_{\text{wts}})^k$ toward the overall runtime. For our intuitive description of the algorithm, we henceforth assume that each guess is exactly correct and that each $\pi^i > 0$.

The next step is to estimate the correlation of the j and j' coordinates, $\text{corr}(j,j') := \mathbf{E}[\mathbf{Z}_j\mathbf{Z}_{j'}]$, for all pairs (j,j') with $1 \leq j < j' \leq n$. Since each distribution $\mathbf{Z}_j\mathbf{Z}_{j'}$ is $\lambda(\epsilon,\delta)$-samplable, we can obtain estimates accurate to within additive error $\pm\epsilon_{\text{matrix}}$ in time poly$(n)\cdot\lambda(\epsilon_{\text{matrix}},\delta)$. For our intuitive description we henceforth assume that each estimated value is exactly correct.

Observe that since $\mathbf{X}_j^i$ and $\mathbf{X}_{j'}^i$ are independent, we have

$$\text{corr}(j,j') = \mathbf{E}[\mathbf{Z}_j\mathbf{Z}_{j'}] = \sum_{i=1}^k \pi^i \mathbf{E}[\mathbf{X}_j^i\mathbf{X}_{j'}^i] = \sum_{i=1}^k \pi^i \mathbf{E}[\mathbf{X}_j^i]\mathbf{E}[\mathbf{X}_{j'}^i] = \sum_{i=1}^k \pi^i \mu_j^i \mu_{j'}^i.$$

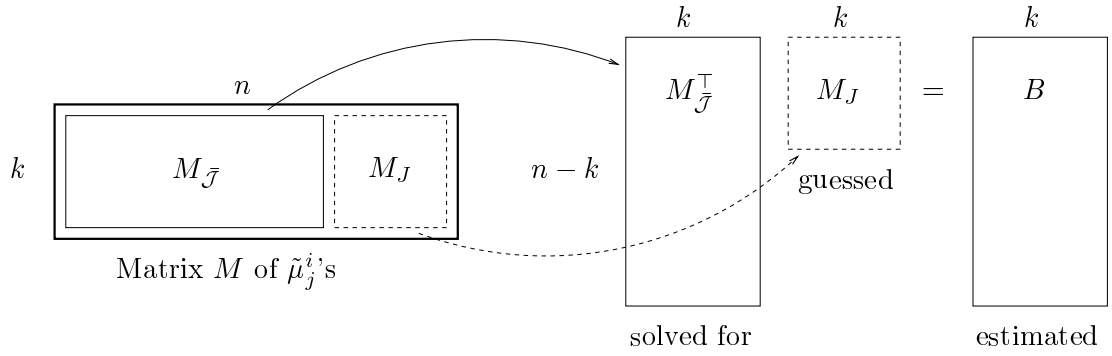

Figure 1: The full rank case. We solve for the unknown $\tilde{\mu}_j^i$ in $M_{\bar{\mathcal{J}}}$.

Let us define

$$\tilde{\mu}_j^i = \sqrt{\pi^i}\mu_j^i$$

and write

$$\tilde{\mu}_j = (\tilde{\mu}_j^1, \tilde{\mu}_j^2, \ldots, \tilde{\mu}_j^k) \in \mathbf{R}^k.$$

We thus have

$$\mathrm{corr}(j, j') = \tilde{\mu}_j \cdot \tilde{\mu}_{j'},$$

where $\cdot$ denotes the dot product in $\mathbf{R}^k$. Hence we assume that Mix-A-Lot has all the pairwise dot products $\tilde{\mu}_j \cdot \tilde{\mu}_{j'}$ for $j \neq j'$.

It remains for Mix-A-Lot to obtain each μ_j^i. Since Mix-A-Lot has each π^i and since each $\pi^i > 0$, it suffices to obtain each $\tilde{\mu}_j^i$. Let M denote the $k \times n$ matrix whose (i, j) entry is the unknown $\tilde{\mu}_j^i$, i.e., the jth column of M is $\tilde{\mu}_j$. Observe that Mix-A-Lot has all of the off-diagonal entries in $M^\top M$, since these are the quantities $\tilde{\mu}_j \cdot \tilde{\mu}_{j'}$ for $j \neq j'$.

We first describe how Mix-A-Lot can obtain all of the entries of M using the off-diagonal entries of $M^\top M$, *assuming that M has full rank*. We will later show how to remove the rank assumption. Assuming M has full rank, there exists some set of k columns of M which are linearly independent, say $J = \{j_1, \ldots, j_k\} \subseteq [n]$. Algorithm Mix-A-Lot *guesses* the set J and *guesses* the vectors $\tilde{\mu}_{j_1}, \ldots, \tilde{\mu}_{j_k}$ to within additive error $\pm\epsilon_{\text{matrix}}$ in each coordinate. Note that the former guess incurs a multiplicative time factor of $\binom{n}{k} \leq n^k$. Since each value $\tilde{\mu}_j^i$ is in the range $[-\mu_{\max}, \mu_{\max}]$, the latter guesses incur a multiplicative time factor of $(\mu_{\max}/\epsilon_{\text{matrix}})^{k^2}$. As usual for this intuitive discussion, we assume that each guessed vector $\tilde{\mu}_{j_1}, \ldots, \tilde{\mu}_{j_k}$ is in fact exactly correct.

Let M_J be the $k \times k$ matrix given by the J-columns of M, and let $M_{\bar{\mathcal{J}}}$ be the $k \times (n-k)$ matrix given by deleting the J-columns of M. Mix-A-Lot now has the entries of M_J, and must compute the remaining unknowns, $M_{\bar{\mathcal{J}}}$. Since Mix-A-Lot has all the off-diagonal entries of $M^\top M$, it has all of the values of $B = M_{\bar{\mathcal{J}}}^\top M_J$. (See Figure 1.) But the columns of M_J are linearly independent, so M_J is invertible and hence Mix-A-Lot can compute $M_{\bar{\mathcal{J}}}^\top = BM_J^{-1}$ in poly(n) time. Mix-A-Lot now has all the entries of $M_{\bar{\mathcal{J}}}$, all the entries of M_J, and the weights π^i, so it is done.

We now turn to the case in which M does not have full rank; say $\mathrm{rank}(M) = k - t$. Algorithm Mix-A-Lot *guesses* this value t (this incurs a multiplicative time factor of $k+1$). Since $\mathrm{rank}(M) = k - t$, there must exist t orthonormal vectors $u_{k-t+1}, \ldots, u_k \in \mathbf{R}^k$ which are orthogonal to all columns of M. (This unnatural indexing will be natural when we give the actual algorithm.) Algorithm Mix-A-Lot *guesses* these t vectors to within error $\pm\epsilon_{\text{matrix}}$ in each coordinate; this incurs a multiplicative time factor of at most $(1/\epsilon_{\text{matrix}})^{k^2}$. As usual, we assume for the intuitive

discussion that each of these guesses is exactly correct. Adjoin these vectors as columns to M, forming M'. The matrix M' has full rank, and MIX-A-LOT knows all the off-diagonal elements of $(M')^\top M'$ (i.e. all the pairwise dot products of M''s columns), since all of the new dot products which involve new columns are simply 0. Thus we can run the algorithm for the full-rank case, and again can compute all of the $\tilde{\mu}_j^i$'s.

Combining all of the multiplicative running time factors from above, the total running time of the algorithm is $\text{poly}(n^k, \lambda(\epsilon_{\text{matrix}}, \delta), (\mu_{\max}/\epsilon_{\text{matrix}})^{k^2}, (1/\epsilon_{\text{wts}})^k)$.

To make this intuitive algorithm rigorous, we need to understand how the errors in our estimates and guesses affect the accuracy of our results when we solve for each $\tilde{\mu}_j^i$ and each μ_j^i. Since we only have approximate values to work with, an aspect of our approach is to use the "essential" rank of M (as measured by its small singular values) instead of its true rank. By working with this essential rank, we can prevent approximation errors from blowing up when we solve the necessary linear systems.

We now state our main result about algorithm MIX-A-LOT. The actual algorithm and proof of correctness are given in Appendices A and B.

Our main theorem describing the performance of MIX-A-LOT is the following:

Theorem 1 *Let $\mathbf{Z}$ be a mixture of product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ with mixing weights $\pi^1, \ldots, \pi^k$ where each $\mu_j^i = \mathbf{E}[\mathbf{X}_j^i]$ satisfies $|\mu_j^i| \leq \mu_{\max}$ and $\mathbf{Z}_j\mathbf{Z}_{j'}$ is $\lambda(\epsilon, \delta)$-samplable for all $j \neq j'$. For any $\epsilon_{\text{wts}}, \epsilon_{\text{means}}, \epsilon_{\text{minwt}} = \text{poly}(\epsilon, 1/n, 1/\mu_{\max})$ where $\epsilon_{\text{wts}} < \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{3/2}/\mu_{\max}$, with probability $1-\delta$ algorithm* MIX-A-LOT *outputs a list of guesses $\langle\{\hat{\pi}^i\}, \{\hat{\mu}_j^i\}\rangle$ such that at least one guess satisfies the following:*

1. *$|\hat{\pi}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all $i = 1 \ldots k$; and*
2. *$|\hat{\mu}_j^i - \mu_j^i| \leq \epsilon_{\text{means}}$ for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$.*

The algorithm runs in time

$$(n\mu_{\max}/\epsilon)^{O(k^3)} \cdot \lambda\left((\epsilon/\mu_{\max}n)^{O(k)}, \delta\right).$$

The proof of Theorem 1 involves a detailed analysis of the singular values of M; see Appendix B.

Remark 1. Note that Theorem 1 guarantees accurate estimation of μ_j^i only for those i such that π^i is not too small. It is easy to see that such a condition on π^i is unavoidable since if π^i is extremely small than a reasonable size sample from $\mathbf{Z}$ will contain no draws from $\mathbf{X}^i$ and thus will give no information about μ_j^i. Note that if π^i is extremely small then it should be possible to ignore $\mathbf{X}^i$ and still obtain an accurate approximation for the overall mixture $\mathbf{Z}$; we make this intuition precise in the following sections.

Remark 2. It is easily seen that algorithm MIX-A-LOT in fact requires only pairwise independence rather than full independence between the coordinates $\mathbf{X}_j^i$ of each component distribution $\mathbf{X}^i$.

4 Bridging the Gap from MIX-A-LOT to Maximum Likelihood

In this section we set up two applications of the algorithm MIX-A-LOT: mixtures of product distributions over $\{0,1\}^n$, and mixtures of n-dimensional axis-aligned Gaussians.

We first note that to use algorithm MIX-A-LOT we must show that the pairwise products of the coordinates of $\mathbf{Z}$ are $\lambda(\epsilon, \delta)$-samplable. We do this separately for both cases below.

Once MIX-A-LOT has been run, we have a list of settings of distribution parameters, at least one of which is accurate. To solve the learning problem, we must first convert each such setting of parameters to an actual mixture of explicit distributions; then we must select one of these candidate mixtures which has small KL divergence from the target distribution. A maximum likelihood algorithm can do this latter step provided that a technical condition holds, namely all candidate mixtures must have pdfs which are bounded above and below (we make this more precise in Section 5).

Thus, to bridge the gap between algorithm MIX-A-LOT and the maximum likelihood algorithm, we need a conversion procedure which has three properties: (i) each candidate parameter setting is converted to an actual mixture distribution; (ii) any accurate setting of parameters is converted into a mixture distribution whose KL divergence from the target is small; (iii) each distribution which the procedure generates has a bounded pdf. In the rest of this section we describe such procedures for mixtures of product distributions over $\{0,1\}^n$ and mixtures of Gaussians.

4.1 Mixtures of product distributions over $\{0,1\}^n$

We now treat the case in which each $\mathbf{X}^i$ is a product distribution over $\{0,1\}^n$, and so $\mathbf{Z}$ is a mixture of product distributions over the Boolean cube. Note that in this case $\mathbf{Z}$ is completely specified by the mixing weights π^i and the expectations $\mu^i_j = \mathbf{E}[\mathbf{X}^i_j]$.

We first observe that each distribution $\mathbf{Z}_j\mathbf{Z}_{j'}$ is a distribution over $\{0,1\}$ and hence is $\lambda(\epsilon,\delta)$-samplable with $\lambda(\epsilon,\delta) = \mathrm{poly}(\frac{1}{\epsilon}, \log\frac{1}{\delta})$. Thus we can apply Theorem 1 with $\mu_{\max} = 1$ and $\lambda(\epsilon,\delta) = \mathrm{poly}(\frac{1}{\epsilon}, \log\frac{1}{\delta})$ and obtain a list of guesses at least one of which is a good parametric estimate for the π^i's and μ^i_j's of $\mathbf{Z}$.

The necessary conversion procedure described above is simple but technical. Here we state a theorem describing the result of executing the conversion procedure on the output list generated by MIX-A-LOT. The conversion procedure is described in Appendix C.2 and the proof of Theorem 2 is given in Appendix C.3.

Theorem 2 *Let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0,1\}^n$. There is a $(n/\epsilon)^{O(k^3)} \cdot \log\frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$, outputs a list of $(n/\epsilon)^{O(k^3)}$ many mixtures of product distributions over $\{0,1\}^n$ with the property that with probability $1-\delta$,*

- *every distribution $\mathbf{Z}'$ in the list satisfies $(\epsilon/6n)^n \leq \mathbf{Z}'(x) \leq 1$ for all $x \in \{0,1\}^n$, and*
- *some distribution $\mathbf{Z}^\star$ in the list satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$.*

4.2 Mixtures of axis-aligned Gaussians

Now we treat the case in which $\mathbf{Z}$ is a mixture of axis-aligned Gaussians $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\mathbf{R}^n$. This continuous setting is somewhat more complicated than the previous discrete setting; we begin by defining some useful notation. As usual we write μ^i_j to denote $\mathbf{E}[\mathbf{X}^i_j]$, and we now write $(\sigma^i_j)^2$ to denote $\mathrm{Var}[\mathbf{X}^i_j]$.

If we expect to learn $\mathbf{Z}$, we will need its component parameters to be reasonable; for example, if the variance in some dimension of a Gaussian in the mixture is not polynomially bounded, then we would not expect to get an accurate estimate of that Gaussian's mean in polynomial time. This motivates the following definition:

Definition 3 *We say that $\mathbf{X}$ is a d-*dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussian *if $\mathbf{X}$ is a d-dimensional axis-aligned Gaussian with the property that each of its one-dimensional coordinate Gaussians $\mathbf{X}_j$ has mean $\mu_j \in [-\mu_{\max}, \mu_{\max}]$ and variance $(\sigma_j)^2 \in [\sigma^2_{\min}, \sigma^2_{\max}]$.*

Throughout this section $\mathbf{Z}$ will be a mixture of n-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians $\mathbf{X}^1, \ldots, \mathbf{X}^k$, where $\mu_{\max}, \sigma^2_{\max} \geq 1$ and $\sigma^2_{\min} \leq 1$, and L will denote $\mu_{\max}\sigma_{\max}/\sigma_{\min}$. Note that $\mathbf{Z}$ is completely specified by the values π^i, μ^i_j, and $(\sigma^i_j)^2$. Our learning algorithm for Gaussians will have a running time that depends polynomially on L; thus, the algorithm is not strongly polynomial.

Finally, in dealing with Gaussians we will need to define $M = M(\theta)$ satisfying

$$\int_{|x|\geq M} \mathbf{X}(x)dx < \theta, \int_{|x|\geq M} |x|\mathbf{X}(x)dx < \theta, \quad \text{and} \quad \int_{|x|\geq M} x^2\mathbf{X}(x)dx < \theta$$

for all one-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians $\mathbf{X}$. This can be achieved with $M = \text{poly}(L, \frac{1}{\theta})$.

4.2.1 Estimating means and variances. Algorithm Mix-A-Lot outputs parametric estimates of mixing weights and means. For Gaussians, we additionally need estimates of *variances*. To achieve this, we run Mix-A-Lot a second time on the random variable $\mathbf{Z}^2$ (i.e. we simply square the value of each coordinate of each draw from $\mathbf{Z}$). This gives us estimates of the mixing weights (again) and also the *second moments*, from which we can recover good estimates of the variances. Having run Mix-A-Lot twice, we essentially take the "cross-product" of the two output lists to obtain a list of candidates, each with mixing weights, means and variances. Proposition 20 (in Appendix D.1) explains this process precisely, and proves that at least one candidate in this new list has (with high probability) good estimates of all the parameters of $\mathbf{Z}$.

4.2.2 Samplability of Gaussians. We will use algorithm Mix-A-Lot to obtain a list of parametric estimates of $\mathbf{Z}$. Each estimate must contain both the means and variances of each component Gaussian. Therefore, we must show that the random variables $\mathbf{Z}_j\mathbf{Z}_{j'}$ are $\lambda(\epsilon, \delta)$-samplable, and also that the random variables $\mathbf{Z}^2_j\mathbf{Z}^2_{j'}$ are $\lambda(\epsilon, \delta)$-samplable. We do this in the following theorem, proved in Appendix D.2:

Proposition 3 *Let $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)$ be a mixture of k two-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians. Then both the random variable $\mathbf{W} := \mathbf{Z}_1\mathbf{Z}_2$ and the random variable $\mathbf{W}^2$ are $\text{poly}(L, \frac{1}{\epsilon}, \log\frac{1}{\delta})$-samplable.*

4.2.3 The conversion procedure. Once again, we must provide a conversion procedure as described earlier. As before the necessary conversion procedure is simple but technical. Here we state the analogous theorem to Theorem 2 for mixtures of Gaussians; the conversion procedure is described in Appendix D.3 and the proof of Theorem 4 is given in Appendix D.4.

Theorem 4 *Let $\mathbf{Z}$ be any unknown mixture of k $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians. Let $M = M(\text{poly}(1/n, 1/L, \epsilon))$. There is a $(Ln/\epsilon)^{O(k^3)} \cdot \log\frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$ outputs a list of $(Ln/\epsilon)^{O(k^3)}$ many mixtures of $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians with the property that with probability $1 - \delta$,*

- *every distribution $\mathbf{Z}'$ in the list satisfies $\exp(-\text{poly}(n, L, 1/\epsilon)) \leq \mathbf{Z}'(x) \leq \text{poly}(L)^n$ for all $x \in [-M, M]^n$, and*
- *some distribution $\mathbf{Z}^\star$ in the list satisfies $\text{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$.*

Note that Theorem 4 guarantees that $\mathbf{Z}'(x)$ has bounded mass only on the range $[-M, M]^n$, whereas the support of $\mathbf{Z}$ goes beyond this range. This issue is addressed in the proof of Theorem 7, where we put together Theorem 4 and the ML procedure.

5 Identifying a Good Distribution Using Maximum Likelihood

Theorems 2 and 4 each give us a list of distributions at least one of which is close to the target distribution we are trying to learn. Now we must *identify* some distribution in the list which is close to the target. In this section we give a simple maximum likelihood algorithm which helps us accomplish this. This is a standard situation (see e.g. Section 4.6 of [11]) and we emphasize that the ideas behind Theorem 5 below are not new. However, we were unable to find in the literature a clear statement of the exact result which we need, so for completeness we give our own statement and proof below.

Let $\mathbf{P}$ be a target distribution over some space X. Let $\mathcal{Q}$ be a set of hypothesis distributions such that at least one $\mathbf{Q}^* \in \mathcal{Q}$ has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^*) \leq \epsilon$. The following algorithm will be used to find a distribution $\mathbf{Q}^{\mathrm{ML}} \in \mathcal{Q}$ which is close to $\mathbf{P}$: Draw a set $\mathcal{S}$ of samples from the distribution $\mathbf{P}$. For each $\mathbf{Q} \in \mathcal{Q}$, compute the log-likelihood

$$\Lambda(\mathbf{Q}) = \sum_{x \in \mathcal{S}} (-\log \mathbf{Q}(x)).$$

Now output the distribution $\mathbf{Q}^{\mathrm{ML}} \in \mathcal{Q}$ such that $\Lambda(\mathbf{Q})$ is minimum. This is known as the Maximum Likelihood (ML) Algorithm since it outputs the distribution in $\mathcal{Q}$ which maximizes $\arg\max_{\mathbf{Q} \in \mathcal{Q}} \prod_{x \in S} \mathbf{Q}(x)$.

We prove Theorem 5 in Appendix E:

Theorem 5 *Let β, α, $\epsilon > 0$ be such that $\alpha < \beta$. Let $\mathcal{Q}$ be a set of hypothesis distributions for some distribution $\mathbf{P}$ over the space X such that at least one $\mathbf{Q}^* \in \mathcal{Q}$ has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^*) \leq \epsilon$. Suppose also that $\alpha \leq \mathbf{Q}(x) \leq \beta$ for all $\mathbf{Q} \in \mathcal{Q}$ and all x such that $\mathbf{P}(x) > 0$.*

Run the ML algorithm on $\mathcal{Q}$ using a set $\mathcal{S}$ of independent samples from $\mathbf{P}$, where $\mathcal{S} = m$. Then, with probability $1 - \delta$, where

$$\delta \quad \leq \quad (|\mathcal{Q}| + 1) \cdot \exp\left(-2m \frac{\epsilon^2}{\log^2(\beta/\alpha)}\right),$$

the algorithm outputs some distribution $\mathbf{Q}^{\mathrm{ML}} \in \mathcal{Q}$ which has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^{\mathrm{ML}}) \leq 4\epsilon$.

6 The Main Learning Results

It is now easy for us to give our first main learning result, for learning mixtures of product distributions over the Boolean cube:

Theorem 6 *Let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0,1\}^n$. There is a $(n/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$ and any $\epsilon, \delta > 0$ as inputs, outputs a mixture $\mathbf{Z}'$ of k product distributions over $\{0,1\}^n$ which with probability at least $1 - \delta$ satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') \leq \epsilon$.*

Proof: Run the algorithm described in Theorem 2. With probability $1 - \delta$ this produces a list of $T = (n/\epsilon)^{O(k^3)}$ hypothesis distributions, one of which has KL divergence at most ϵ from $\mathbf{Z}$ and all of which put weight at least $(\epsilon/6n)^n$ on every point in $\{0,1\}^n$. Now run the ML algorithm with $\alpha = (\epsilon/6n)^n$, $\beta = 1$, and $m = \mathrm{poly}(n, 1/\epsilon) \ln(T/\delta)$. By Theorem 5, with probability at least $1 - \delta$ it outputs a hypothesis with KL divergence at most 4ϵ from $\mathbf{Z}$. Thus with overall probability $1 - 2\delta$ we get a hypothesis with KL divergence at most 4ϵ from $\mathbf{Z}$, and the total running time is $(n/\epsilon)^{O(k^3)} \cdot \log(1/\delta)$. Replacing ϵ by $\epsilon/4$ and δ by $\delta/2$ we are done. ■

A little more work is required for our second main result, on learning mixtures of Gaussians.

Theorem 7 *Let $\mathbf{Z}$ be any unknown mixture of k n-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians. There is a $(Ln/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$ and any $\epsilon, \delta > 0$ as inputs, outputs a mixture $\mathbf{Z}'$ of k ($(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded) Gaussians which with probability at least $1 - \delta$ satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') \leq \epsilon$.*

Proof: Run the algorithm given by Theorem 4. With probability $1 - \delta$ this produces a list of $T = (Ln/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ hypothesis distributions, one of which, $\mathbf{Z}^\star$, has KL divergence at most ϵ from $\mathbf{Z}$ and all of which have their pdfs bounded between $\exp(-\mathrm{poly}(n, L, 1/\epsilon))$ and $\mathrm{poly}(L)^n$ for all $x \in [-M, M]^n$.

We now consider $\mathbf{Z}_M$, the M-truncated version of $\mathbf{Z}$; this is simply the distribution obtained by restricting the support of $\mathbf{Z}$ to be $[-M, M]^n$ and scaling so that $\mathbf{Z}_M$ is a distribution. We prove the following proposition in Appendix F:

Proposition 8 *Let $\mathbf{P}$ and $\mathbf{Q}$ be any mixtures of n-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians. Let $\mathbf{P}_M$ denote the M-truncated version of $\mathbf{P}$, where M is chosen as in Theorem 4. Then we have $|\mathrm{KL}(\mathbf{P}_M||\mathbf{Q}) - \mathrm{KL}(\mathbf{P}||\mathbf{Q})| \leq 4\epsilon + 2\epsilon \cdot \mathrm{KL}(\mathbf{P}||\mathbf{Q})$.*

This proposition implies that $\mathrm{KL}(\mathbf{Z}_M||\mathbf{Z}^\star) \leq 7\epsilon$.

Now run the ML algorithm with $m = \mathrm{poly}(n, L, 1/\epsilon) \log(M/\delta)$ on this list of hypothesis distributions *using $\mathbf{Z}_M$ as the target distribution.* (We can obtain draws from $\mathbf{Z}_M$ using rejection sampling from $\mathbf{Z}$; with probability $1 - \delta$ this incurs only a negligible increase in the time required to obtain m draws.) Note that running the algorithm with $\mathbf{Z}_M$ as the target distribution lets us assert that all hypothesis distributions have pdfs bounded above and below on the support of the target distribution, as is required by Theorem 5. (In contrast, since the support of $\mathbf{Z}$ is all of $\mathbf{R}^n$, we cannot guarantee that our hypothesis distributions have pdf bounds on the support of $\mathbf{Z}$.) By Theorem 5, with probability at least $1 - \delta$ the ML algorithm outputs a hypothesis $\mathbf{Z}^{\mathrm{ML}}$ which satisfies $\mathrm{KL}(\mathbf{Z}_M||\mathbf{Z}^{\mathrm{ML}}) \leq 28\epsilon$.

It remains only to bound $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^{\mathrm{ML}})$. By Proposition 8 we have

$$\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^{\mathrm{ML}}) \leq 28\epsilon + 4\epsilon + 2\epsilon \mathrm{KL}(\mathbf{Z}||\mathbf{Z}^{\mathrm{ML}})$$

which implies that $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^{\mathrm{ML}}) \leq 33\epsilon$. The running time of the overall algorithm is $(Ln/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ and the theorem is proved. ■

7 Hardness of Learning Mixtures of Product Distributions

In this section we give evidence that the class of mixtures of $k(n)$ product distributions over the Boolean cube may be hard to learn in polynomial time for any $k(n) = \omega(1)$.

Before describing our results, we recall some standard terminology about Boolean *decision trees.* Recall that a decision tree is a rooted binary tree in which each internal node has two children and is labeled with a variable and each leaf is labeled with a bit $b \in \{-1, +1\}$. A decision tree T computes a Boolean function $f : \{0,1\}^n \to \{-1, 1\}$ in the obvious way: on input $x \in \{0,1\}^n$, if variable x_i is at the root of T we go to either the left or right subtree depending on whether x_i is 0 or 1. Continue in this fashion until reaching a bit leaf; the value of this bit is $f(x)$.

Our main result in this section is the following theorem proved in Appendix G:

Theorem 9 *For any function $k(n)$, if there is a poly(n) time algorithm which learns a mixture of $k(n)$ product distributions over $\{0,1\}^n$, then there is a poly(n) time uniform distribution PAC learning algorithm which learns the class of all $k(n)$-leaf decision trees.*

We note that after years of intensive research, no poly(n) time uniform distribution PAC learning algorithm is known which can learn $k(n)$-leaf decision trees for any $k(n) = \omega(1)$; indeed, such an algorithm would be a major breakthrough in computational learning theory.[1] The fastest algorithms to date [10, 2] can learn $k(n)$-leaf decision trees under the uniform distribution in time $n^{\log k(n)}$.

8 Conclusions

We have shown how to learn mixtures of product distributions over $\{0,1\}^n$ and axis-aligned Gaussians in polynomial time. The methods we use are quite general; we believe that they can be used to learn mixtures of many other types of multivariate product distributions which are definable in terms of their moments. (Of course, other technical conditions must hold, such as requiring samplability.) For example, one should be able to adapt our methods to learn mixtures of products of exponential distributions or beta distributions.

It is natural to ask if our methods can be improved to learn mixtures of distributions which are not necessarily product distributions on $\mathbf{R}^n$. In particular, is it possible to learn non-axis-aligned Gaussians efficiently in our model? Note that our techniques only require that that the coordinate distributions be pairwise independent.

Finally, one may ask if it is possible to improve the efficiency of our learning algorithms — can the running times be reduced to $n^{O(k^2)}$, to $n^{O(k)}$, or even $n^{O(\log k)}$?

References

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

[2] A. Blum. Rank-r decision trees are a subclass of r-decision lists. *Information Processing Letters*, 42(4):183–185, 1992.

[3] A. Blum. Learning a function of r relevant variables (open problem). In *Proceedings of the 16th Annual Conference on Learning Theory and 7th Kernel Workshop*, pages 731–733, 2003.

[4] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of the Twenty-Sixth Annual Symposium on Theory of Computing*, pages 253–262, 1994.

[5] T. Cover and J. Thomas. *Elements of Information Theory*. Wiley, 1991.

[6] M. Cryan, L. Goldberg, and P. Goldberg. Evolutionary trees can be learned in polynomial time in the two state general Markov model. *SIAM Journal on Computing*, 31(2):375–397, 2002.

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

[8] 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.

[1] Avrim Blum has offered a \$1000 prize for solving a subproblem of the $k(n) = n$ case and a \$500 prize for a subproblem of the $k(n) = \log n$ case; see [3].

[9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistics Soc. Ser. B*, 39:1–38, 1977.

[10] A. Ehrenfeucht and D. Haussler. Learning decision trees from random examples. *Information and Computation*, 82(3):231–246, 1989.

[11] 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.

[12] S. Goreinov, E. Tyrtyshnikov, and N. Zamarashkin. A theory of pseudoskeleton approximations. *Linear Algebra and its Applications*, 261:1–21, 1997.

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

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

[15] M. Kearns, Y. Mansour, D. Ron, R. Rubinfeld, R. Schapire, and L. Sellie. On the learnability of discrete distributions. In *Proceedings of the Twenty-Sixth Symposium on Theory of Computing*, pages 273–282, 1994.

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

[17] S. Newcomb. A generalized theory of the combination of observations so as to obtain the best result. *Amer. J. Math.*, 8:343–366, 1886.

[18] K. Pearson. Contribution to the mathematical theory of evolution. *Phil. Trans. Roy. Soc. A*, 185:71–110, 1894.

[19] A. Ray. Personal communication, 2003.

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

[21] M. Seeger. *Bayesian Gaussian Process Models: PAC-Bayesian Generalization Error Bounds and Sparse Approximations*. PhD thesis.

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

[23] 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.

A Algorithm Mix-A-Lot

Algorithm Mix-A-Lot has access to samples from the mixture $\mathbf{Z}$ and takes as input parameters $\epsilon_{\text{wts}}, \epsilon_{\text{means}}, \epsilon_{\text{minwt}} < 1$, $\mu_{\max} \geq 1$ where:

- ϵ_{wts} is the desired accuracy for each mixing weight π^i;

- ϵ_{means} is the desired accuracy for those μ_j^i's which have $\pi^i \geq \epsilon_{\text{minwt}}$;
- $\epsilon_{\text{wts}} \leq \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{3/2}/\mu_{\max}$; and
- $\mu_{\max} \geq 1$ and $|\mu_j^i| \leq \mu_{\max}$ for all $j \neq j'$.

Algorithm MIX-A-LOT:

1. Make guesses for the mixing weights $\hat{\pi}^1, \ldots, \hat{\pi}^k \in [0,1]$ to within $\pm\epsilon_{\text{wts}}$. If s of the weights are guessed to be smaller than $\epsilon_{\text{minwt}} - \epsilon_{\text{wts}}$, eliminate them and treat k as $k-s$ in what follows.
2. Make empirical estimates $\widehat{\text{corr}}(j,j')$ for all correlations $\text{corr}(j,j') = \mathbf{E}[\mathbf{Z}_j\mathbf{Z}_{j'}]$ $= \tilde{\mu}_j \cdot \tilde{\mu}_{j'}$ for $j \neq j'$ to within $\pm\epsilon_{\text{matrix}}$.
3. Let M be the $k \times n$ matrix of unknowns $(M_{ij}) = (\tilde{\mu}_j^i)$, and guess an integer $0 \leq t \leq k$ (the essential rank-deficiency of M).
4. Guess t vectors $\hat{u}_{k-t+1}, \ldots, \hat{u}_k \in [-1,1]^k$ to within $\pm\epsilon_{\text{matrix}}$ in each coordinate and augment M with these as columns, forming $\widehat{M}'$.
5. Guess a subset of exactly k column indices of $\widehat{M}'$; write these indices as $\mathcal{J} = J \cup J'$, where J corresponds to columns from the original matrix M and J' corresponds to augmented columns. Make guesses $\{\hat{\tilde{\mu}}_j^i : i \in [k], j \in J\}$ for the entries of M in columns J to within $\pm\epsilon_{\text{matrix}}$ (where each guess covers the range $[-\mu_{\max}, \mu_{\max}]$). Let $\widehat{M}'_{\mathcal{J}}$ denote the matrix of guesses for all the columns in $\mathcal{J}$. (See Figure 2.)
6. Let $\bar{\mathcal{J}}$ denote the columns of M other than J, and let $M_{\bar{\mathcal{J}}}$ denote the matrix of remaining unknowns formed by these columns. Let $\widehat{B}$ be the matrix with rows indexed by $\bar{\mathcal{J}}$ and columns indexed by $\mathcal{J}$ whose (j,j') entry is the estimate $\widehat{\text{corr}}(j,j')$ of $\tilde{\mu}_j \cdot \tilde{\mu}_{j'}$ if $j' \in J$ or is 0 if $j' \in J'$. Using the entries of $\hat{B}$ and $\widehat{M}'_{\mathcal{J}}$ (all of which are known), solve the system $M_{\bar{\mathcal{J}}}^{\top}\widehat{M}'_{\mathcal{J}} = \widehat{B}$ to obtain estimates $\hat{\tilde{\mu}}_j^i$ for the entries of $M_{\bar{\mathcal{J}}}$ (which are the unknown $\tilde{\mu}_j^i$'s), thus producing estimates $\hat{\tilde{\mu}}_j^i$ for all entries of M. (If the matrix $\widehat{M}'_{\mathcal{J}}$ is singular, simply abandon the current guess.)
7. From the estimated values $\hat{\tilde{\mu}}_j^i$, compute the estimates $\hat{\mu}_j^i = \hat{\tilde{\mu}}_j^i/\sqrt{\hat{\pi}^i}$ for all i,j. (Note that $\hat{\pi}^i$ is never 0 since $\epsilon_{\text{minwt}} > \epsilon_{\text{wts}}$.)
8. Output the guesses $\{\hat{\pi}^i\}$ and $\{\hat{\mu}_j^i\}$.

Our main theorem describing the performance of MIX-A-LOT is the following:

Theorem 1: *Let $\mathbf{Z}$ be a mixture of product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ with mixing weights $\pi^1, \ldots, \pi^k$ where each $\mu_j^i = \mathbf{E}[\mathbf{X}_j^i]$ satisfies $|\mu_j^i| \leq \mu_{\max}$ and $\mathbf{Z}_j\mathbf{Z}_{j'}$ is $\lambda(\epsilon,\delta)$-samplable for all $j \neq j'$. Let $\epsilon_{\text{wts}}, \epsilon_{\text{means}}, \epsilon_{\text{minwt}} < 1 \leq \mu_{\max}$ be such that $\epsilon_{\text{wts}} \leq \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{3/2}/\mu_{\max}$. Let $\tilde{\epsilon} = \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{1/2}/2$, let $\tau = \tilde{\epsilon}/(12\mu_{\max}^2 k^{5/2}(kn+1))$, and let $\epsilon_{\text{matrix}} = \tilde{\epsilon}\tau^k$. With probability $1-\delta$ algorithm* MIX-A-LOT *outputs a list of guesses $\langle\{\hat{\pi}^i\}, \{\hat{\mu}_j^i\}\rangle$ such that at least one guess satisfies the following:*

1. *$|\hat{\pi}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all $i = 1 \ldots k$; and*

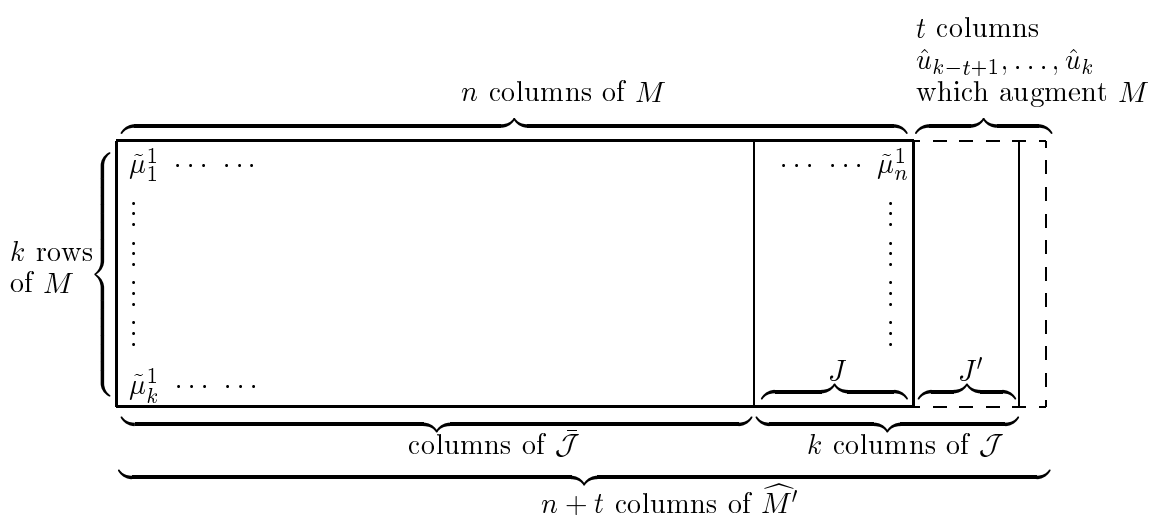

Figure 2: A depiction of the matrix used by Mix-A-Lot. For ease of illustration the columns J of M are depicted as being the rightmost columns of M, and the columns J' from the augmenting columns $\hat{u}_{k-t+1}, \ldots, \hat{u}_k$ are depicted as being the leftmost of those columns.

2. $|\hat{\mu}^i_j - \mu^i_j| \leq \epsilon_{\text{means}}$ *for all* i, j *such that* $\pi^i \geq \epsilon_{\text{minwt}}$.

The algorithm runs in time

$$\left(\frac{n\mu_{\max}}{\epsilon_{\text{means}}\epsilon_{\text{minwt}}}\right)^{O(k^3)} \lambda\left((\epsilon_{\text{means}}\epsilon_{\text{minwt}}/(n\mu_{\max}))^{O(k)}, \delta\right) \cdot \left(\frac{1}{\epsilon_{\text{wts}}}\right)^{O(k)}.$$

B Proof of Theorem 1

The proof of Theorem 1 requires several concepts and facts from linear algebra. We review these in Section B.1 and give the proof of Theorem 1 in Section B.2.

B.1 Linear algebra preliminaries

Let $A = (a_{ij})$ be any $k \times n$ real matrix. Let $\sigma_1 \geq \cdots \geq \sigma_k \geq 0$ be the singular values of A, and let $u_1, \ldots, u_k$ be the corresponding left singular vectors of A (i.e. the columns of U where $A = U\Sigma V$). Recall that

- the vectors $u_1, \ldots, u_k$ form an orthonormal basis for $\mathbf{R}^k$;
- $\sigma_1 = \max_{\|x\|_2=1} \|x^\top A\|_2$ and $\sigma_k = \min_{\|x\|_2=1} \|x^\top A\|_2$.

The *Frobenius norm* $\|A\|_F$ of a $k \times n$ matrix A is defined as $\|A\|_F = \sqrt{\sum_{i,j}(A_{i,j})^2}$. Recall that $\sigma_k(A)$ equals the Frobenius norm distance from the $k \times n$ matrix A to the nearest rank-deficient matrix $\tilde{A}$, i.e.

$$\sigma_k(A) = \min_{rank(\tilde{A})<k} \|A - \tilde{A}\|_F.$$

The *spectral norm* $\|A\|_2$ of a $k \times n$ matrix A is $\|A\|_2 = \max_{\|x\|=1} \|Ax\|$. It is well known that $\|A\|_2 = \sigma_1$ and $\|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_k^2}$; note that this implies $\|A\|_2 \leq \|A\|_F$.

We will several times use the following proposition which we prove in Appendix B.3:

Proposition 10 *Let A be a $k \times n$ real matrix with $\sigma_k(A) \geq \epsilon$. Then there exists a subset of columns $J \subseteq [n]$ with $|J| = k$ such that $\sigma_k(A_J) \geq \epsilon/\sqrt{k(n-k)+1}$.*

The quantity $\sigma_k(A)$ is useful in bounding the perturbation in the solution of a perturbed linear system. We will also use several times the following theorem which we prove in Appendix B.4:

Theorem 11 *Let A be a nonsingular $k \times k$ matrix, b be a k-dimensional vector, and x the solution to $Ax = b$. Suppose A' is a $k \times k$ matrix such that each entry of $A - A'$ is at most ϵ_{matrix} in magnitude, where $k^2\epsilon_{\text{matrix}} < \sigma_k(A)$. Let b' be a k-dimensional vector satisfying $\|b - b'\|_\infty \leq \epsilon_{\text{rhs}}$ and let x' be the solution to $A'x' = b'$. Then we have*

$$\|x - x'\|_\infty \leq \frac{\epsilon_{\text{matrix}} k^{5/2} \|x\|_\infty + \epsilon_{\text{rhs}} k^{1/2}}{\sigma_k(A) - \epsilon_{\text{matrix}} k^2}.$$

In particular, if $K \geq \max\{\|x\|_\infty, 1\}$ and $\epsilon_{\text{matrix}} k^2 \leq \sigma_k(A)/2$, then

$$\|x - x'\|_\infty \leq 4k^{5/2} K \, \frac{\epsilon_{\text{matrix}} + \epsilon_{\text{rhs}}}{\sigma_k(A)}.$$

B.2 Proof of Theorem 1

Proof: First we analyze the running time of Mix-A-Lot. As described in Section 3.2, the overall runtime of this algorithm is easily seen to be $\text{poly}(n^k, \lambda(\epsilon_{\text{matrix}}, \delta), (\mu_{\max}/\epsilon_{\text{matrix}})^{k^2}, (1/\epsilon_{\text{wts}})^k)$ Since we have that $\epsilon_{\text{matrix}} = (\epsilon_{\text{means}}\epsilon_{\text{minwt}}/nk\mu_{\max})^{O(k)}$, this gives the claimed time bound.

Now we prove correctness by showing that some guess has the claimed properties.

In Step 1 there will be some accurate guess for the mixing weights $\hat{\pi}^1, \ldots, \hat{\pi}^k$ such that $|\hat{\pi}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all i. For this guess, note that in the second part of Step 1 the algorithm will not eliminate any product distribution $\mathbf{X}^i$ whose mixing weight π^i is at least ϵ_{minwt}. Since we make no claim about the accuracy of $\hat{\mu}_j^i$ for those i which have $\pi^i < \epsilon_{\text{minwt}}$, we can ignore those i's and assume for the rest of this proof that $\pi^i \geq \epsilon_{\text{minwt}}$ and $\hat{\pi}^i \geq \epsilon_{\text{minwt}} - \epsilon_{\text{wts}} > 0$ for all i.

Since each $\mathbf{Z}_j\mathbf{Z}_{j'}$ is $\lambda(\epsilon, \delta)$-samplable, by the discussion in Section 3.1 we can assume that all empirical estimates in Step 2 will be accurate to within $\pm\epsilon_{\text{matrix}}$.

To analyze Steps 3–6 we must consider various cases depending on the singular values of M. In each case we will show that the estimates produced for all $\tilde{\mu}_j^i$'s are accurate to within an additive $\pm\tilde{\epsilon}$.

Case 1: $\sigma_1(M) \leq \tilde{\epsilon}$. Since $\sigma_1(M)$ is at least as large as the magnitude of the largest entry of M, in this case we have $|\tilde{\mu}_j^i| \leq \tilde{\epsilon}$ for all i, j. In Step 3 the algorithm will guess $t = k$ and in Step 4 the algorithm will guess $\hat{u}_1, \ldots, \hat{u}_k$ to be exactly the standard basis vectors in $\mathbf{R}^k$ (as long as $1/\epsilon_{\text{matrix}}$ is an integer, which we may take to be the case without loss of generality). In Step 5 the algorithm will guess $\mathcal{J} = J'$ to be $\{\hat{u}_1, \ldots, \hat{u}_k\}$ and consequently $\widehat{M'}_{\mathcal{J}}$ will be the $k \times k$ identity matrix. In Step 6 the matrix B will have all entries 0 so the algorithm will produce the estimate $\hat{\tilde{\mu}}_j^i = 0$ for each $\tilde{\mu}_j^i$. But $|\tilde{\mu}_j^i| \leq \tilde{\epsilon}$ for all i, j, so these estimates are correct to within an additive $\tilde{\epsilon}$, as desired.

Case 2a: $\sigma_1(M) > \tilde{\epsilon}$ **and** $\sigma_k(M) \geq \epsilon_{\text{matrix}}/\tau$. In this case, in Step 3 the algorithm will guess $t = 0$ and Step 4 will be vacuous. Since $\sigma_k(M) \geq \epsilon_{\text{matrix}}/\tau$, by Proposition 10 there must exist some

set of k columns $\mathcal{J} = J$ such that $\sigma_k(M_{\mathcal{J}}) \geq \epsilon_{\text{matrix}}/\tau\sqrt{k(n-k)+1}$. In Step 5 the algorithm will guess these columns and will guess their entries to within additive error $\pm\epsilon_{\text{matrix}}$ (note that each true entry lies in $[-\mu_{\max}, \mu_{\max}]$). To analyze the estimates $\hat{\tilde{\mu}}_j^i$ obtained in Step 6 we use Theorem 11. We have $\mu_{\max} \geq \max\{\|\tilde{\mu}_j\|_\infty, 1\}$, and by the definition of τ we have $\epsilon_{\text{matrix}}k^2 \leq (1/2)\epsilon_{\text{matrix}}/\tau\sqrt{k(n-k)+1} \leq \sigma_k(M_{\mathcal{J}})/2$. Hence by the second part of the theorem, the additive error in the estimates $\hat{\tilde{\mu}}_j^i$ produced in Step 6 is at most

$$4k^{5/2}\mu_{\max}\,\frac{\epsilon_{\text{matrix}}+\epsilon_{\text{matrix}}}{\epsilon_{\text{matrix}}/\tau\sqrt{k(n-k)+1}} = 8k^{5/2}\sqrt{k(n-k)+1}\cdot\mu_{\max}\tau < \tilde{\epsilon},$$

as desired (the last inequality follows easily from the definition of τ).

Case 2b: $\sigma_1(M) > \tilde{\epsilon}$ and $\sigma_k(M) < \epsilon_{\text{matrix}}/\tau$. This is the last and most complicated case. Since in this case $\sigma_1(M) > \tilde{\epsilon}$ and $\sigma_k(M) < \tilde{\epsilon}\tau^{k-1}$ (by definition of ϵ_{matrix}), it follows that there must be some $0 < t < k$ such that

$$\frac{\sigma_{k-t+1}(M)}{\sigma_{k-t}(M)} \leq \tau. \tag{1}$$

If we take the largest such value t we must in addition have

$$\sigma_{k-t}(M) \geq \sigma_1(M)\tau^{k-t-1} \geq \tilde{\epsilon}\tau^{k-t-1} > \tilde{\epsilon}\tau^{k-1} = \epsilon_{\text{matrix}}/\tau. \tag{2}$$

In Step 3 the algorithm will guess this t. Let $u_{k-t+1}, \ldots, u_k$ be the left singular vectors of M corresponding to its smallest t singular values. In Step 4 the algorithm will make guesses $\hat{u}_{k-t+1}, \ldots, \hat{u}_k$ for each of these vectors which are accurate to within $\pm\epsilon_{\text{matrix}}$ in each coordinate. Let M' denote the matrix M with the *true* singular vectors $u_{k-t+1}, \ldots, u_k$ adjoined as columns. We have the following proposition which we prove in Appendix B.5:

Proposition 12 *Let M be a $k \times n$ matrix and let $u_{k-t+1}, \ldots, u_k$ and M' be as described above. Then we have*

$$\sigma_k(M') \geq \min\{1, \sigma_{k-t}(M)\}, \tag{3}$$

and for all $k-t+1 \leq \ell \leq k$ and for all columns $\tilde{\mu}_1, \ldots \tilde{\mu}_n$ of M we have

$$|\tilde{\mu}_j \cdot u_\ell| \leq \sigma_{k-t+1}(M). \tag{4}$$

Applying Proposition 10 (noting that M' may have up to $n+k$ columns), we may conclude that there is some subset $\mathcal{J}$ of M''s columns with $|\mathcal{J}| = k$ such that

$$\sigma_k(M'_{\mathcal{J}}) \geq \sigma_k(M')/\sqrt{kn+1} \geq \min\{1, \sigma_{k-t}(M)\}/\sqrt{kn+1}. \tag{5}$$

By (3) and (2) we have that $\sigma_k(M') \geq \epsilon_{\text{matrix}}/\tau$, so we also have

$$\sigma_k(M'_{\mathcal{J}}) \geq \epsilon_{\text{matrix}}/\tau\sqrt{kn+1}. \tag{6}$$

Write $\mathcal{J} = J \cup J'$ where J are the columns from M and J' are the columns from the augmented u_ℓ's. In Step 5, there will be a correct guess for $\mathcal{J}$ and accurate guesses $\hat{\tilde{\mu}}_j^i$ for all $i \in [k]$ and $j \in J$ to within $\pm\epsilon_{\text{matrix}}$ of the true values $\tilde{\mu}_j^i$. Note that the accurately guessed entries of $\widehat{M'}_{\mathcal{J}}$ are all within $\pm\epsilon_{\text{matrix}}$ of the true entries $M'_{\mathcal{J}}$.

Let $\bar{\mathcal{J}}$ be the columns of M other than J and let B be the true matrix $M_{\bar{\mathcal{J}}}^\top M'_{\mathcal{J}}$. Consider any row B_j, $j \in \bar{\mathcal{J}}$, whose columns (entries) are indexed by indices $j' \in \mathcal{J}$. For indices $j' \in J$ we have $B_{j,j'} = \tilde{\mu}_j \cdot \tilde{\mu}_{j'}$; hence the estimate $\hat{B}_{j,j'} = \widehat{\text{corr}}(j, j')$ in Step 6 of the algorithm is

accurate to within $\pm\epsilon_{\text{matrix}}$. For indices $j' \in J'$ we have $|B_{j,j'}| = |\tilde{\mu}_j \cdot u_{j'}| \leq \sigma_{k-t+1}(M)$ by (4). Hence the estimate $\widehat{B}_{j,j'} = 0$ is accurate to within $\pm\sigma_{k-t+1}(M)$. Thus we conclude that for every row $j \in \bar{\mathcal{J}}$, $\|\widehat{B}_j - B_j\|_\infty \leq \max\{\epsilon_{\text{matrix}}, \sigma_{k-t+1}(M)\} \leq \epsilon_{\text{matrix}} + \tau\sigma_{k-t}(M)$, where the last step uses (1). As before we use Theorem 11 to bound the error of the estimates $\hat{\tilde{\mu}}_j^i$ obtained in Step 6. As in the analysis in Case 2a, $\mu_{\max} \geq \max\{\|\tilde{\mu}_j\|_\infty, 1\}$, and the definition of τ gives $\epsilon_{\text{matrix}}k^2 \leq (1/2)\epsilon_{\text{matrix}}/\tau\sqrt{kn+1} \leq \sigma_k(M'_{\mathcal{J}})/2$ (where the last inequality uses (6)). Hence by the last part of Theorem 11 the additive error of the estimates is at most

$$\begin{aligned} 4k^{5/2}\mu_{\max}\,\frac{\epsilon_{\text{matrix}} + \epsilon_{\text{matrix}} + \tau\sigma_{k-t}(M)}{\sigma_k(M'_{\mathcal{J}})} &= 4k^{5/2}\mu_{\max}\,\frac{2\epsilon_{\text{matrix}} + \tau\sigma_{k-t}(M)}{\sigma_k(M'_{\mathcal{J}})} \\ &\leq 4k^{5/2}\mu_{\max}\left[2\tau\sqrt{kn+1} + \frac{\tau\sigma_{k-t}(M)}{\sigma_k(M'_{\mathcal{J}})}\right] \qquad (7) \end{aligned}$$

where the inequality uses (6).

We now have two cases depending on the value of $\min\{1, \sigma_{k-t}(M)\}$. If $\sigma_{k-t}(M) \leq 1$ then by (5) we have $\tau\sigma_{k-t}(M)/\sigma_k(M'_{\mathcal{J}}) \leq \tau\sqrt{kn+1}$ and hence (7) is at most $12k^{5/2}\mu_{\max}\tau\sqrt{kn+1}$ which is easily seen to be at most $\tilde{\epsilon}$ by the definition of τ. On the other hand, if $\sigma_{k-t}(M) > 1$ then by (5) we have $\sqrt{kn+1} \geq 1/\sigma_k(M'_{\mathcal{J}})$ and hence

$$\tau\sigma_{k-t}(M)/\sigma_k(M'_{\mathcal{J}}) \leq \tau\sigma_{k-t}(M)\sqrt{kn+1} \leq \tau\mu_{\max}\sqrt{kn}\sqrt{kn+1}$$

where the last (crude) inequality holds since $\sigma_{k-t}(M) \leq \sigma_1(M) = \|M\|_2 \leq \|M\|_F$ which is at most $\sqrt{kn}\mu_{\max}$ since each of the kn entries of M is at most $\mu_{\max}$ in magnitude. We thus have that (7) is at most

$$4k^{5/2}\mu_{\max}\left[2\tau\sqrt{kn+1} + \tau\mu_{\max}\sqrt{kn}\sqrt{kn+1}\right] \leq 4k^{5/2}\mu_{\max}[2\tau\mu_{\max}(kn+1)] < \tilde{\epsilon}$$

as desired, where the last inequality follows from our definition of τ.

We have now completed all the cases and shown that in every case, the algorithm produces estimates $\hat{\tilde{\mu}}_j^i$ at the end of Step 6 that are accurate to within an additive $\tilde{\epsilon}$.

Finally we consider Step 7 of the algorithm. We have guesses for all π^i which are accurate to within $\pm\epsilon_{\text{wts}}$ and we have guesses for all $\tilde{\mu}_j^i$ accurate to within $\pm\tilde{\epsilon}$. Since the function $g(x,y) = y/\sqrt{x}$ satisfies

$$\sup_{\substack{x\in[\epsilon_{\text{minwt}},1]\\ y\in[-\mu_{\max},\mu_{\max}]}} \left|\frac{\partial}{\partial x}g(x,y)\right| = \mu_{\max}/2\epsilon_{\text{minwt}}^{3/2} \quad \text{and} \quad \sup_{\substack{x\in[\epsilon_{\text{minwt}},1]\\ y\in[-\mu_{\max},\mu_{\max}]}} \left|\frac{\partial}{\partial y}g(x,y)\right| < 1/\epsilon_{\text{minwt}}^{1/2},$$

the Mean Value Theorem implies that in Step 7 we obtain guesses $\hat{\mu}_j^i$ for all μ_j^i which are accurate to within additive error

$$\epsilon_{\text{wts}}\mu_{\max}/2\epsilon_{\text{minwt}}^{3/2} + \tilde{\epsilon}/\epsilon_{\text{minwt}}^{1/2}.$$

By the definition of $\tilde{\epsilon}$ and the fact that $\epsilon_{\text{wts}} \leq \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{3/2}/\mu_{\max}$, both summands are at most $\epsilon_{\text{means}}/2$ and the proof is complete. ■

B.3 Proof of Proposition 10

Recall Proposition 10:

Proposition 10: *Let A be a $k \times n$ real matrix with $\sigma_k(A) \geq \epsilon$. Then there exists a subset of columns $J \subseteq [n]$ with $|J| = k$ such that $\sigma_k(A_J) \geq \epsilon/\sqrt{k(n-k)+1}$.*

(We note that a much weaker version of Proposition 10 follows easily from the Cauchy-Binet theorem, which states that the squared volume of A equals the sum of the squared volumes of A's $k \times k$ submatrices.)

Our proof of Proposition 10 uses the following result due to Goreinov *et al.* [12]. For completeness we give their simple proof below.

Theorem 13 [12] *Let M be a $k \times n$ real matrix with orthonormal rows. Then there is a $k \times k$ submatrix M_J which has $\sigma_k(M_J) \geq 1/\sqrt{k(n-k)+1}$.*

Proof: For a $k \times k$ matrix Q let $\mathrm{Vol}(Q)$ denote $\prod_{i=1}^{k} \sigma_i(Q)$. Without loss of generality we may assume that the $k \times k$ submatrix P of M which has maximum volume over all $k \times k$ submatrices is obtained by taking columns $1, \ldots, k$ of M. Since $\mathrm{Vol}(Q) = |\det(Q)|$, it follows that the $k \times k$ submatrix of maximum volume in the matrix

$$\tilde{M} \equiv P^{-1}M = [I\ T]$$

is the identity matrix located in columns $1, \ldots, k$. This implies that $|\tilde{M}_{i,j}| \leq 1$ for all i, j; for if $|\tilde{M}_{i,j}| > 1$ then by swapping columns i and j the first k columns would give a submatrix whose volume is greater than 1. Hence each entry of the $k \times (n-k)$ matrix T has magnitude at most 1, and we have

$$\frac{1}{\sigma_k(P)} = \sigma_1(P^{-1}) = \sigma_1(\tilde{M}) = \|\tilde{M}\|_2 \leq \sqrt{\|I\|_2^2 + \|T\|_2^2} \leq \sqrt{\|T\|_F^2 + 1} \leq \sqrt{k(n-k)+1}.$$

where the first equality is a standard fact, the second holds since the rows of M are orthonormal, and the first inequality follows from the definition of $\|M\|_2$. This proves the theorem taking $M_J = P$. ■

Proof of Proposition 10: By the singular value decomposition we have $A = U\Sigma V$ where U is a $k \times k$ matrix with orthonormal columns, Σ is a $k \times k$ diagonal matrix with diagonal entries $\sigma_1, \ldots, \sigma_k$, and V is a $k \times n$ matrix with orthonormal rows. Let V_J be the $k \times k$ submatrix of V whose existence is asserted by Theorem 13, so $\sigma_k(V_J) \geq 1/\sqrt{k(n-k)+1}$. We have $\sigma_k(U) = 1$ (since the rows of U are orthonormal) and $\sigma_k(\Sigma) = \sigma_k$, so

$$\sigma_k(U\Sigma V_J) \geq \sigma_k(U)\sigma_k(\Sigma)\sigma_k(V_J) \geq \sigma_k/\sqrt{k(n-k)+1}$$

where the inequality holds since $\sigma_k(PQ) \geq \sigma_k(P)\sigma_k(Q)$ for any $k \times k$ matrices P, Q (this is easily seen from the variational characterization $\sigma_k(P) = \min_{\|x\|=1} \|Px\|$.) The theorem follows by observing that $U\Sigma V_J$ is the $k \times k$ submatrix of A whose columns are in J. ■

B.4 Proof of Theorem 11

Recall Theorem 11:

Theorem 11: *Let A be a nonsingular $k \times k$ matrix, b be a k-dimensional vector, and x the solution to $Ax = b$. Suppose A' is a $k \times k$ matrix such that each entry of $A - A'$ is at most $\epsilon_{\mathrm{matrix}}$ in magnitude, where $k^2\epsilon_{\mathrm{matrix}} < \sigma_k(A)$. Let b' be a k-dimensional vector satisfying $\|b - b'\|_\infty \leq \epsilon_{\mathrm{rhs}}$ and let x' be the solution to $A'x' = b'$. Then we have*

$$\|x - x'\|_\infty \leq \frac{\epsilon_{\mathrm{matrix}} k^{5/2} \|x\|_\infty + \epsilon_{\mathrm{rhs}} k^{1/2}}{\sigma_k(A) - \epsilon_{\mathrm{matrix}} k^2}.$$

Theorem 11 follows directly from the following theorem by comparing the L_2 and L_∞ norms.

Theorem 14 *Let A be a nonsingular $k \times k$ matrix, b be a k-dimensional vector, and x the solution to $Ax = b$. Suppose A' is a $k \times k$ matrix satisfying $\|A - A'\|_F \leq \epsilon_1 < \sigma_k(A)$. Let b' be a k-dimensional vector satisfying $\|b - b'\|_2 \leq \epsilon_2$ and let x' be the solution to $A'x' = b'$. Then*

$$\|x - x'\|_2 \leq \frac{\epsilon_1 \|x\|_2 + \epsilon_2}{\sigma_k(A) - \epsilon_1}.$$

Proof of Theorem 14: We first note that since $\|A - A'\|_F \leq \epsilon_1 < \sigma_k$, we have $\text{rank}(A') = k$ so the vector x' is well defined.

Let $E = A - A'$ and $\eta = b - b'$. We have

$$\begin{aligned} x &= A^{-1}b = A^{-1}(b' + \eta) = A^{-1}A'x' + A^{-1}\eta = (I - A^{-1}E)x' + A^{-1}\eta \\ &= x' - A^{-1}Ex' + A^{-1}\eta. \end{aligned}$$

Consequently we have

$$\begin{aligned} \|x - x'\|_2 &= \|A^{-1}Ex' - A^{-1}\eta\|_2 \\ &\leq \|A^{-1}Ex'\|_2 + \|A^{-1}\eta\|_2 \\ &\leq \|A^{-1}\|_2 \|E\|_2 \|x'\|_2 + \|A^{-1}\|_2 \|\eta\|_2 \\ &\leq \|A^{-1}\|_2 (\|E\|_F \|x'\|_2 + \|\eta\|_2) \\ &\leq \|A^{-1}\|_2 \left(\epsilon_1 \|x'\|_2 + \epsilon_2\right). \end{aligned}$$

Since $\|A^{-1}\|_2 = \sigma_1(A^{-1}) = 1/\sigma_k(A)$ we have

$$\|x - x'\|_2 \leq \frac{1}{\sigma_k(A)} \left(\epsilon_1 \|x'\|_2 + \epsilon_2\right) \leq \frac{1}{\sigma_k(A)} \left(\epsilon_1 \|x - x'\|_2 + \epsilon_1 \|x\|_2 + \epsilon_2\right)$$

from which the theorem follows. ■

B.5 Proof of Proposition 12

Recall Proposition 12:

Proposition 12: *Let M be a $k \times n$ matrix with columns $\tilde{\mu}_1, \ldots, \tilde{\mu}_n$. Let $u_{k-t+1}, \ldots, u_k$ be the left singular vectors corresponding to the smallest singular values $\sigma_{k-t+1}, \ldots, \sigma_k$ of M. Let M' be M with the vectors $u_{k-t+1}, \ldots, u_k$ adjoined as columns. Then*

$$\sigma_k(M') \geq \min\{1, \sigma_{k-t}(M)\},$$

and for all $k - t + 1 \leq \ell \leq k$ and for all columns $\tilde{\mu}_1, \ldots \tilde{\mu}_n$ of M we have

$$|\tilde{\mu}_j \cdot u_\ell| \leq \sigma_{k-t+1}(M).$$

Proof: Recall that the singular value decomposition gives us $M = U\Sigma V$ where U is a $k \times k$ matrix with orthonormal columns $u_1, \ldots, u_k$; Σ is a $k \times k$ diagonal matrix with diagonal elements $\sigma_1 \geq \cdots \geq \sigma_k \geq 0$; and V is a $k \times n$ matrix with orthonormal rows. It follows that for any vector $x \in \mathbf{R}^k$ we have

$$\|x^\top M\|^2 = \sigma_1^2 (x^\top u_1)^2 + \cdots + \sigma_k^2 (x^\top u_k)^2.$$

Let R denote the $k \times t$ matrix whose columns are $u_{k-t+1}, \ldots, u_k$, so we have $M' = [M\ R]$. It is easily verified that the left singular vectors of R are simply $u_{k-t+1}, \ldots, u_k$, while the singular values of R are all 1. Consequently we have

$$\|x^\top R\|^2 = (x^\top u_{k-t+1})^2 + \cdots + (x^\top u_k)^2$$

for any $x \in \mathbf{R}^k$.

Now recall the variational characterization of $\sigma_k(M')$, namely $\sigma_k(M') = \min_{\|x\|=1} \|x^\top M'\|$. Since $\|x^\top M'\| = \sqrt{\|x^\top M\|^2 + \|x^\top R\|^2}$, we have

$$\sigma_k(M') = \min_{\|x\|=1} \sqrt{\sigma_1^2(x^\top u_1)^2 + \cdots + \sigma_k^2(x^\top u_k)^2 + (x^\top u_{k-t+1})^2 + \cdots + (x^\top u_k)^2}. \tag{8}$$

Since $u_1, \ldots, u_k$ form an orthonormal basis for $\mathbf{R}^k$ we have that $(x^\top u_1)^2 + \cdots + (x^\top u_k)^2 = 1$ for all $\|x\| = 1$. If we let $\alpha_x = (x^\top u_{k-t+1})^2 + \cdots + (x^\top u_k)^2$, then the quantity inside the square root of (8) is at least $\sigma_{k-t}^2(1-\alpha_x) + \alpha_x$. This proves the first inequality of the proposition.

For the second inequality, we observe that $\tilde{\mu}_j \cdot u_\ell = u_\ell^\top U \Sigma v_j$ where v_j is the j-th column of V. Since U is orthonormal and $\Sigma_{\ell,\ell} = \sigma_\ell$, we thus have

$$|u_\ell^\top U \Sigma v_j| = |\sigma_\ell v_{\ell,j}| \leq \sigma_\ell \leq \sigma_{k-t+1}$$

where the first inequality holds since the rows of V are orthonormal and hence each entry of V must be at most 1 in magnitude. ■

C Processing candidate mixtures of $\{0,1\}$ product distributions

In this section we consider $\mathbf{Z}$ to be an unknown mixture of product distributions over $\{0,1\}^n$. Our goal is to prove Theorem 2, which allows us to take a list of candidates output by Mix-A-Lot, and convert it into a list of candidates that meets the conditions of the ML procedure.

C.1 Some useful propositions

Here we give some useful elementary propositions which will be used in the proofs of Theorem 2 and its supporting claims.

Proposition 15 *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *be discrete probability distributions satisfying* $\|\mathbf{P} - \mathbf{Q}\|_\infty \leq \epsilon$ *and* $\mathbf{Q} \geq \alpha$. *Then* $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \leq \epsilon/\alpha$.

Proof: Using the elementary inequality $|\ln a - \ln b| \leq \frac{|a-b|}{\min\{a,b\}}$ which holds for all $a, b \in [0,1]$, we have $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) = \sum_{x \in \mathrm{supp}(\mathbf{P})} \mathbf{P}(x)(\ln \mathbf{P}(x) - \ln \mathbf{Q}(x)) \leq \sum \mathbf{P}(x) \frac{\epsilon}{\min\{\mathbf{P}(x), \mathbf{Q}(x)\}} \leq \epsilon/\alpha$. ■

Proposition 16 *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *denote distributions over* $\{0,1\}$ *with means* p *and* q *respectively. Suppose* $|p-q| < \epsilon$. *Let* $0 < \alpha < 1/2$. *Let* $\mathbf{Q}'$ *denote another distribution over* $\{0,1\}$ *with mean* q', *where*

$$q' = \begin{cases} \alpha & \text{if } q < \alpha \\ 1-\alpha & \text{if } q > 1-\alpha \\ q & \text{o.w.} \end{cases}$$

Then, $\mathrm{KL}(\mathbf{P}||\mathbf{Q}') \leq \max\{\epsilon/\alpha, 2\alpha\}$.

Proof: Suppose $|p-q'| > |p-q|$. Then, $q \neq q'$ and so either $q = \alpha$, or $q = 1-\alpha$. In the first case, $q' > q$, and so $p \leq q < q' = \alpha$. In the second case, $q' < q$, and so $p \geq q > q' = 1-\alpha$. Therefore, at least one of the following hold: (i) $|p-q'| \leq \epsilon$, or (ii) $q' = \alpha$ and $p \leq q'$, or (iii) $q' = 1-\alpha$ and $p \geq q'$. In (i), since $\mathbf{Q}'$ has weight at least α on both 0 and 1, we have by Proposition 15, $\mathrm{KL}(\mathbf{P}||\mathbf{Q}') \leq \epsilon/\alpha$. In (ii) and (iii), we have $\mathrm{KL}(\mathbf{P}||\mathbf{Q}') \leq \ln \frac{1}{1-\alpha} \leq 2\alpha$. ■

Proposition 17 *Suppose* $\mathbf{P}_1, \ldots, \mathbf{P}_n$ *and* $\mathbf{Q}_1, \ldots, \mathbf{Q}_n$ *are distributions satisfying* $\mathrm{KL}(\mathbf{P}_i||\mathbf{Q}_i) \leq \epsilon_i$ *for all* i. *Then* $\mathrm{KL}(\mathbf{P}_1 \times \cdots \times \mathbf{P}_n||\mathbf{Q}_1 \times \cdots \times \mathbf{Q}_n) \leq \sum_{i=1}^n \epsilon_i$.

Proof: We prove the case $n = 2$:

$$\begin{aligned}
\mathrm{KL}(\mathbf{P}_1 \times \mathbf{P}_2||\mathbf{Q}_1 \times \mathbf{Q}_2) &= \iint \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_1(x)\mathbf{P}_2(y)}{\mathbf{Q}_1(x)\mathbf{Q}_2(y)} dxdy \\
&= \iint \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_1(x)}{\mathbf{Q}_1(x)} dxdy + \iint \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_2(y)}{\mathbf{Q}_2(y)} dxdy \\
&= \int \mathbf{P}_2(y)\mathrm{KL}(\mathbf{P}_1||\mathbf{Q}_1)dy + \int \mathbf{P}_1(x)\mathrm{KL}(\mathbf{P}_2||\mathbf{Q}_2)dx \\
&\leq \epsilon_1 + \epsilon_2.
\end{aligned}$$

The general case follows by induction. ■

Proposition 18 *Suppose* $\pi^1, \ldots, \pi^k$, $\gamma^1, \ldots, \gamma^k$ *are mixing weights satisfying* $\sum \pi^i = \sum \gamma^i = 1$, $|\pi^i - \gamma^i| \leq \epsilon_1$ *for all* i, *and* $\gamma^i \geq \epsilon_2$ *for all* i. *Let* $\mathcal{I} = \{i : \pi^i \geq \epsilon_3\}$. *Suppose* $\mathbf{P}^1, \ldots, \mathbf{P}^k$ *and* $\mathbf{Q}^1, \ldots, \mathbf{Q}^k$ *are distributions where* $\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) \leq \epsilon$ *for all* $i \in \mathcal{I}$. *Then, letting* $\mathbf{P}$ *denote the* π*-mixture of the* $\mathbf{P}^i$*'s and* $\mathbf{Q}$ *the* γ*-mixture of the* $\mathbf{Q}^i$*'s, for any* $\epsilon_4 > \epsilon_1$ *we have*

$$\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \leq \epsilon + k\epsilon_3 \cdot \max_i \mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) + k\epsilon_4 \ln \frac{\epsilon_4}{\epsilon_2} + \frac{\epsilon_1}{\epsilon_4 - \epsilon_1}.$$

Proof:

$$\begin{aligned}
\mathrm{KL}(\mathbf{P}||\mathbf{Q}) &= \int \Big(\sum_i \pi^i \mathbf{P}^i\Big) \ln \frac{\sum_i \pi^i \mathbf{P}^i}{\sum_i \gamma^i \mathbf{Q}^i} \\
&\leq \int \sum_i \pi^i \mathbf{P}^i \ln \frac{\pi^i \mathbf{P}^i}{\gamma^i \mathbf{Q}^i} \qquad \text{(by the log-sum inequality [5])} \\
&= \sum_i \pi^i \int \Big(\mathbf{P}^i \ln \frac{\mathbf{P}^i}{\mathbf{Q}^i} + \mathbf{P}^i \ln \frac{\pi^i}{\gamma^i}\Big) \\
&= \sum_i \pi^i \mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) + \sum_i \pi^i \ln \frac{\pi^i}{\gamma^i} \\
&= \left(\sum_{i \in \mathcal{I}} \pi^i \mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i)\right) + \left(\sum_{i \notin \mathcal{I}} \pi^i \mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i)\right) + \sum_i \pi^i \ln \frac{\pi^i}{\gamma^i}. \qquad (9)
\end{aligned}$$

For the first term of (9), we have

$$\sum_{i \in \mathcal{I}} \pi^i \mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) \leq \epsilon.$$

For the second term of (9), we have

$$\sum_{i \notin \mathcal{I}} \pi^i \mathrm{KL}(\mathbf{P}^i || \mathbf{Q}^i) \quad \leq \quad k\epsilon_3 \cdot \max_i \{\mathrm{KL}(\mathbf{P}^i || \mathbf{Q}^i)\}.$$

For the third term of (9), letting $\mathcal{I}' = \{i \in \mathcal{I} : \pi^i \geq \epsilon_4\}$, we have

$$\sum_i \pi^i \ln \frac{\pi^i}{\gamma^i} = \sum_{i \notin \mathcal{I}'} \pi^i \ln \frac{\pi^i}{\gamma^i} + \sum_{i \in \mathcal{I}'} \pi^i \ln \frac{\pi^i}{\gamma^i}. \tag{10}$$

For the first sum in (10) we have

$$\sum_{i \notin \mathcal{I}'} \pi^i \ln \frac{\pi^i}{\gamma^i} \leq k\epsilon_4 \ln \frac{\epsilon_4}{\epsilon_2}.$$

Since $\gamma^i \geq \pi^i - \epsilon_1$ for all i, we have that for all $i \in \mathcal{I}'$

$$\frac{\pi^i}{\gamma^i} \geq \frac{\epsilon_4}{\epsilon_4 - \epsilon_1} = 1 + \frac{\epsilon_1}{\epsilon_4 - \epsilon_1}.$$

Hence for the second sum in (10), we have

$$\sum_{i \in \mathcal{I}'} \pi^i \ln \frac{\pi^i}{\gamma^i} \leq \sum_{i \in \mathcal{I}'} \pi^i \ln \left(1 + \frac{\epsilon_1}{\epsilon_4 - \epsilon_1}\right) \leq \frac{\epsilon_1}{\epsilon_4 - \epsilon_1}.$$

Putting all the bounds together the proof is done. ■

C.2 Processing the candidates

The following theorem defines a process that converts a single estimate for the π^i's and μ^i_j's of $\mathbf{Z}$ to a mixture of product distributions over the cube that has minimum mass on every point in $\{0,1\}^n$, as required by the ML procedure. In addition, it guarantees that if the parametric estimate was accurate (close to the true parameters of $\mathbf{Z}$), then the process outputs a distribution with small KL divergence relative to $\mathbf{Z}$. This will be a key step in the proof of Theorem 2.

Theorem 19 *There is a simple efficient procedure $\mathcal{A}$ which takes values ϵ_{means} and $\hat{\pi}^i, \hat{\mu}^i_j$ as inputs and outputs a mixture $\dot{\mathbf{Z}}$ of k product distributions over $\{0,1\}^n$ with mixing weights $\dot{\pi}^i$ and means $\dot{\mu}^i_j$ satisfying*

- $\sum_{i=1}^k \dot{\pi}^i = 1$, *and*
- $\alpha_{\text{cube}} := \epsilon_{\text{means}}^{n/2} \leq \dot{\mathbf{Z}}(x) \leq 1$ *for all* $x \in \{0,1\}^n$.

Furthermore, suppose $\mathbf{Z}$ is a mixture of k product distributions on $\{0,1\}^n$ with mixing weights $\pi^1, \ldots, \pi^k$ and means μ^i_j, and that the following are satisfied:

- *for $i = 1 \ldots k$ we have $|\pi^i - \hat{\pi}^i| \leq \epsilon_{\text{wts}}$, and*
- *for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$ we have $|\mu^i_j - \hat{\mu}^i_j| \leq \epsilon_{\text{means}}$.*

Then $\dot{\mathbf{Z}}$ will satisfy

$$\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq \eta_{\mathrm{cube}}(\epsilon_{\mathrm{means}}, \epsilon_{\mathrm{wts}}, \epsilon_{\mathrm{minwt}}),$$

where

$$\eta_{\mathrm{cube}}(\epsilon_{\mathrm{means}}, \epsilon_{\mathrm{wts}}, \epsilon_{\mathrm{minwt}}) := n \cdot (2\epsilon_{\mathrm{means}}^{1/2}) + k\epsilon_{\mathrm{minwt}} n \left(\ln 2 + \frac{1}{2} \ln \frac{1}{\epsilon_{\mathrm{means}}} \right) + \epsilon_{\mathrm{wts}}^{1/3}.$$

Proof: We construct a mixture $\dot{\mathbf{Z}}$ of product distributions $\dot{\mathbf{Z}}^1, \ldots, \dot{\mathbf{Z}}^k$ by defining new mixing weights $\dot{\pi}^i$ and expectations $\dot{\mu}_j^i$. The procedure $\mathcal{A}$ is defined as follows:

1. Let $\alpha = \epsilon_{\mathrm{means}}^{1/2}$. For all i, j, set

$$\dot{\mu}_j^i = \begin{cases} \alpha & \text{if } \hat{\mu}_j^i < \alpha \\ 1 - \alpha & \text{if } \hat{\mu}_j^i > 1 - \alpha \\ \hat{\mu}_j^i & \text{o.w.} \end{cases}$$

2. For all $i = 1, \ldots, k$ let

$$\ddot{\pi}^i = \begin{cases} \hat{\pi}^i & \text{if } \hat{\pi}^i \geq \epsilon_{\mathrm{wts}} \\ \epsilon_{\mathrm{wts}} & \text{if } \hat{\pi}^i < \epsilon_{\mathrm{wts}} \end{cases}$$

   Now let s be such that $s \sum_{i=1}^k \ddot{\pi}^i = 1$, and take $\dot{\pi}^i = s\ddot{\pi}^i$.

Consider a particular μ_j^i and $\dot{\mu}_j^i$ where i is such that $\pi^i \geq \epsilon_{\mathrm{minwt}}$. Let $\mathbf{P}$ and $\mathbf{Q}$ denote the distributions over $\{0, 1\}$ with means μ_j^i and $\dot{\mu}_j^i$, respectively. By Proposition 16, we have $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \leq \max\{\epsilon_{\mathrm{means}}/\alpha, 2\alpha\} = 2\epsilon_{\mathrm{means}}^{1/2}$. Each $\mathbf{Z}^i$ and $\dot{\mathbf{Z}}^i$ is the product of n such simple distributions over $\{0, 1\}$. Therefore, by Proposition 17, we have $\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i) \leq n \cdot (2\epsilon_{\mathrm{means}}^{1/2})$ for all i with $\pi^i \geq \epsilon_{\mathrm{minwt}}$.

We clearly have $\dot{\mathbf{Z}}(x) \leq 1$ for all x. By construction, we also have $\dot{\mathbf{Z}}^i(x) \geq \alpha^n$ for all $x \in \{0, 1\}^n$, and hence $\dot{\mathbf{Z}}(x) \geq \alpha_{\mathrm{cube}} := \alpha^n$ for all x. It follows that for all $i = 1, \ldots, k$ we have

$$\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i) \leq H(\mathbf{Z}^i) + \ln \frac{1}{\alpha^n} \leq n(\ln 2 + \ln \frac{1}{\alpha}).$$

Our goal is to apply Proposition 18 to bound $\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}})$; to satisfy the conditions of Proposition 18 we must upper bound $|\pi^i - \dot{\pi}^i|$ and lower bound $\dot{\pi}^i$ for all i. We now do this.

If $\hat{\pi}^i \geq \epsilon_{\mathrm{wts}}$ then we have $\ddot{\pi}^i = \hat{\pi}^i$ so $|\pi^i - \ddot{\pi}^i| \leq \epsilon_{\mathrm{wts}}$. On the other hand, if $\hat{\pi}^i < \epsilon_{\mathrm{wts}}$ then it must be the case that $\pi^i \leq 2\epsilon_{\mathrm{wts}}$ so we again have $|\pi^i - \ddot{\pi}^i| \leq \epsilon_{\mathrm{wts}}$. Since $\sum_{i=1}^k \pi^i = 1$ it follows that

$$\left| \sum_{i=1}^k \ddot{\pi}^i - 1 \right| \leq k\epsilon_{\mathrm{wts}} \tag{11}$$

and thus

$$\sum_{i=1}^k \ddot{\pi}^i \in [1 - k\epsilon_{\mathrm{wts}}, 1 + k\epsilon_{\mathrm{wts}}].$$

By definition of s this gives

$$s \in \left[\frac{1}{1 + k\epsilon_{\mathrm{wts}}}, \frac{1}{1 - k\epsilon_{\mathrm{wts}}} \right].$$

Multiplying inequality (11) by s and recalling that $s\sum_{i=1}^{k}\ddot{\pi}^i = 1$, and $\epsilon_{\text{wts}} \leq 1/(2k)$, we obtain

$$|1-s| \leq sk\epsilon_{\text{wts}} \leq \frac{k\epsilon_{\text{wts}}}{1-k\epsilon_{\text{wts}}} \leq 2k\epsilon_{\text{wts}}.$$

Thus, we have

$$\begin{aligned} |\pi^i - \dot{\pi}^i| &\leq |\pi^i - \ddot{\pi}^i| + |\ddot{\pi}^i - \dot{\pi}^i| \\ &\leq \epsilon_{\text{wts}} + |\ddot{\pi}^i - \dot{\pi}^i| \\ &= \epsilon_{\text{wts}} + |(1-s)\ddot{\pi}^i| \\ &\leq \epsilon_{\text{wts}} + 2k\epsilon_{\text{wts}}|\ddot{\pi}^i| \\ &\leq \epsilon_{\text{wts}} + 2k\epsilon_{\text{wts}}; \end{aligned}$$

certainly, this gives $|\pi^i - \dot{\pi}^i| \leq 3k\epsilon_{\text{wts}}$. To lower bound $\dot{\pi}^i$, we note that since $\ddot{\pi}^i \geq \epsilon_{\text{wts}}$ for all i, we have

$$\dot{\pi}^i = s\ddot{\pi}^i \geq \frac{1}{1+k\epsilon_{\text{wts}}}\ddot{\pi}^i \geq \frac{\epsilon_{\text{wts}}}{1+k\epsilon_{\text{wts}}} \geq \frac{\epsilon_{\text{wts}}}{2}.$$

We are now ready to apply Proposition 18 with the following parameter settings:

$$\epsilon_1 = 3k\epsilon_{\text{wts}};\ \ \epsilon_2 = \epsilon_{\text{wts}}/2;\ \ \epsilon_3 = \epsilon_{\text{minwt}};\ \ \epsilon_4 = \epsilon_{\text{wts}}^{1/2};\quad \epsilon = n\cdot(2\epsilon_{\text{means}}^{1/2}).$$

Proposition 18 implies:

$$\begin{aligned} \text{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq\ & n\cdot(2\epsilon_{\text{means}}^{1/2}) + k\epsilon_{\text{minwt}} n\left(\ln 2 + \frac{1}{2}\ln\frac{1}{\epsilon_{\text{means}}}\right) + \\ & \left[k\epsilon_{\text{wts}}^{1/2}\ln\frac{\epsilon_{\text{wts}}^{1/2}}{\epsilon_{\text{wts}}/2} + \frac{3k\epsilon_{\text{wts}}}{\epsilon_{\text{wts}}^{1/2} - 3k\epsilon_{\text{wts}}}\right]. \end{aligned}$$

Considering the terms of the expression in brackets above, we have that

$$k\epsilon_{\text{wts}}^{1/2}\ln\frac{\epsilon_{\text{wts}}^{1/2}}{\epsilon_{\text{wts}}/2} = k\epsilon_{\text{wts}}^{1/2}\ln\frac{2}{\epsilon_{\text{wts}}^{1/2}} \leq \frac{1}{2}\epsilon_{\text{wts}}^{1/3}$$

and

$$\frac{3k\epsilon_{\text{wts}}}{\epsilon_{\text{wts}}^{1/2} - 3k\epsilon_{\text{wts}}} \leq 6k\epsilon_{\text{wts}}^{1/2} \leq \frac{1}{2}\epsilon_{\text{wts}}^{1/3}.$$

Hence

$$\text{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq n\cdot(2\epsilon_{\text{means}}^{1/2}) + k\epsilon_{\text{minwt}} n\left(\ln 2 + \frac{1}{2}\ln\frac{1}{\epsilon_{\text{means}}}\right) + \epsilon_{\text{wts}}^{1/3}.$$

■

C.3 Proof of Theorem 2

With all the tools of the previous sections, we are now ready to prove Theorem 2:

Theorem 2: *Let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0,1\}^n$. There is a $(n/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$, outputs a list of $(n/\epsilon)^{O(k^3)}$ many mixtures of product distributions over $\{0,1\}^n$ with the property that with probability $1-\delta$,*

- *every distribution $\mathbf{Z}'$ in the list satisfies $(\epsilon/6n)^n \leq \mathbf{Z}'(x) \leq 1$ for all $x \in \{0,1\}^n$, and*
- *some distribution $\mathbf{Z}^\star$ in the list satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$.*

Proof: Let $\epsilon, \delta > 0$ be given. Run Mix-A-Lot with $\epsilon_{\text{means}} = \frac{\epsilon^2}{36n^2}, \epsilon_{\text{minwt}} = \frac{\epsilon^2}{5kn^2}, \epsilon_{\text{wts}} = \epsilon^5/1000k^2n^5$, and $\mu_{\max} = 1$ (note that $\epsilon_{\text{wts}} < \epsilon_{\text{means}}\epsilon_{\text{minwt}}^{3/2}/\mu_{\max}$ as required by Theorem 1). This takes time $(n/\epsilon)^{O(k^3)} \log(1/\delta)$. We get as output $(n/\epsilon)^{O(k^3)}$ many candidate parameter settings $\langle \{\hat{\pi}^i\}, \{\hat{\mu}_j^i\} \rangle$ with the guarantee that with probability $1-\delta$ at least one of the settings satisfies

- for $i = 1 \ldots k$ we have $|\pi^i - \hat{\pi}^i| \leq \epsilon_{\text{wts}}$, and
- for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$ we have $|\mu_j^i - \hat{\mu}_j^i| \leq \epsilon_{\text{means}}$.

We now pass all of these settings through Theorem 19. It follows that the resulting distributions each satisfy $\epsilon_{\text{means}}^{n/2} = (\epsilon/6n)^n \leq \mathbf{Z}'(x) \leq 1$ for all $x \in \{0,1\}^n$, and one can check that with our setting of parameters $\eta_{\text{cube}} \leq \epsilon$, so that one of the resulting distributions $\mathbf{Z}^\star$ satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$. ■

D Processing candidate mixtures of Gaussians

In this section, we will prove theorems necessary to use algorithm Mix-A-Lot on mixtures of Gaussians, and to apply Mix-A-Lot to learn mixtures of Gaussians.

Throughout this section $\mathbf{Z}$ will be a mixture of n-dimensional $(\mu_{\max}, \sigma_{\min}^2, \sigma_{\max}^2)$-bounded Gaussians $\mathbf{X}^1, \ldots, \mathbf{X}^k$, where $\mu_{\max}, \sigma_{\max}^2 \geq 1$ and $\sigma_{\min}^2 \leq 1$, and L will denote $\mu_{\max}\sigma_{\max}/\sigma_{\min}$.

D.1 Using Mix-A-Lot on both means and variances

In this section we show how to use Mix-A-Lot twice to obtain a list of parametric estimates of $\mathbf{Z}$, one of which is accurate for all the parameters of $\mathbf{Z}$:

Proposition 20 *Let $\mathbf{Z}$ be a mixture of n-dimensional $(\mu_{\max}, \sigma_{\min}, \sigma_{\max})$-bounded Gaussians $\mathbf{X}^1, \ldots, \mathbf{X}^k$ with mixing weights $\pi^1, \ldots, \pi^k$, means μ_j^i and variances $(\sigma_j^i)^2$. Then there is an algorithm running in time*

$$\left(\frac{1}{\epsilon_{\text{wts}}}\right)^{O(k)} \cdot \left(\frac{n\mu_{\max}\sigma_{\max}}{\epsilon_{\text{means}}\epsilon_{\text{minwt}}}\right)^{O(k^3)} \cdot \log\left(\frac{1}{\delta}\right)$$

which, with probability at least $1-\delta$, outputs a list of $\left(\frac{1}{\epsilon_{\text{wts}}}\right)^{O(k)} \cdot \left(\frac{n\mu_{\max}}{\epsilon_{\text{means}}\epsilon_{\text{vars}}\epsilon_{\text{minwt}}}\right)^{O(k^3)}$ many guesses $\langle \{\hat{\pi}^i\}, \{\hat{\mu}_j^i\}, \{(\hat{\sigma}_j^i)^2\} \rangle$ such that at least one guess satisfies the following:

1. *$|\hat{\pi}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all $i = 1 \ldots k$; and*
2. *$|\hat{\mu}_j^i - \mu_j^i| \leq \epsilon_{\text{means}}$ and $|(\hat{\sigma}_j^i)^2 - (\sigma_j^i)^2| \leq \epsilon_{\text{vars}} := 2\epsilon_{\text{means}}$ for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$.*

Proof: Run the algorithm MIX-A-LOT with the random variable $\mathbf{Z}$, taking the parameter "δ" to be $\delta/2$. By Proposition 3 (proved in the next subsection) this takes at most the claimed running time. MIX-A-LOT outputs a list List1 of candidates for the mixing weights and expectations, List1 $= [\ldots, \langle\hat{\pi}^i, \hat{\mu}^i_j\rangle, \ldots]$, which with probability at least $1 - \delta/2$ has a "good" entry which satisfies

1. $|\hat{\pi}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all $i = 1 \ldots k$; and
2. $|\hat{\mu}^i_j - \mu^i_j| \leq \epsilon_{\text{means}}$ for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$.

Define $(s^i_j)^2 = \mathbf{E}[(\mathbf{X}^i_j)^2] = (\sigma^i_j)^2 + (\mu^i_j)^2$. Run the algorithm MIX-A-LOT again on the squared random variable $\mathbf{Z}^2$, with "$\mu_{\max}$" $= \sigma^2_{\max} + \mu^2_{\max}$ and "δ" $= \delta/2$. By Proposition 3, this again takes at most the claimed running time. This time MIX-A-LOT outputs a list List2 of candidates for the mixing weights (again) and second moments, List2 $= [\ldots, \langle\hat{\hat{\pi}}^i, (\hat{s}^i_j)^2\rangle, \ldots]$, which with probability at least $1 - \delta/2$ has a "good" entry which satisfies

1. $|\hat{\hat{\pi}}^i - \pi^i| \leq \epsilon_{\text{wts}}$ for all $i = 1 \ldots k$; and
2. $|(\hat{s}^i_j)^2 - (s^i_j)^2| \leq \epsilon_{\text{means}}$ for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$.

We now form the "cross product" of the two lists. (Again, this can be done in the claimed running time.) Specifically, for each candidate $\langle\hat{\pi}^i, \hat{\mu}^i_j\rangle$ in List1, we create a new candidate using every possible candidate $\langle\hat{\hat{\pi}}^i, (\hat{s}^i_j)^2\rangle$ in List2 by forming $\langle\hat{\pi}^i, \hat{\mu}^i_j, (\hat{\sigma}^i_j)^2 := (\hat{s}^i_j)^2 - \hat{\mu}^i_j\rangle$ (we discard $\hat{\hat{\pi}}^i$). Note that when we have the "good" candidate from List1 matched with the "good" candidate from List2, the resulting candidate indeed satisfies all of the conclusions of the theorem (the error in $(\hat{\sigma}^i_j)^2$ is at most $\epsilon_{\text{means}} + \epsilon_{\text{means}} = \epsilon_{\text{vars}}$ from the triangle inequality). ■

D.2 Proof of Proposition 3: Samplability

Recall Proposition 3:

Proposition 3: *Let $\mathbf{Z} = (\mathbf{Z}_1, \mathbf{Z}_2)$ be a mixture of k two-dimensional $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians. Then both the random variable $\mathbf{W} := \mathbf{Z}_1\mathbf{Z}_2$ and the random variable $\mathbf{W}^2$ are $\text{poly}(L, \frac{1}{\epsilon}, \log\frac{1}{\delta})$-samplable.*

Proof: We shall prove the proposition for $\mathbf{W}^2$; the proof for $\mathbf{W}$ is similar but slightly simpler.

Let the mixing weights be $\pi^1, \ldots, \pi^k$ and suppose that $\mathbf{Z}_j$ is a mixture of $\mathbf{X}^1_j, \ldots, \mathbf{X}^k_j$ for $j = 1, 2$. Let $s = \mathbf{E}[\mathbf{W}^2]$.

Recall the quantity $M = M(\theta)$ and take $C = M^4 = \text{poly}(L/\theta)$. Let $\mathbf{W}^2_C$ denote the random variable $\mathbf{W}^2$ conditioned on the event $|\mathbf{W}^2| \leq C$. Observe that

$$\Pr[\mathbf{W}^2 > C] = \Pr[\mathbf{W}^2 > M^4] \leq \Pr[|\mathbf{Z}_1| > M] + \Pr[|\mathbf{Z}_2| > M] \leq 2\theta, \tag{12}$$

using the fact that $\mathbf{Z}_1$ and $\mathbf{Z}_2$ are $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$-bounded Gaussians and the definition of M.

We shall show that $|\mathbf{E}[\mathbf{W}^2_C] - s| \leq \epsilon/2$. Our sampling algorithm for $\mathbf{W}^2$ will be to sample from $\mathbf{W}^2_C$ using rejection sampling and to compute and output the empirical mean of $\mathbf{W}^2_C$. Since the random variable $\mathbf{W}^2_C$ is bounded in the range $[-C, C]$, by the Hoeffding bound if we take $\text{poly}(C/\epsilon, \log(1/\delta)) = \text{poly}(L/\epsilon\theta, \log(1/\delta))$ samples from $\mathbf{W}^2_C$ then with probability $1 - \delta$ the empirical mean of $\mathbf{W}^2_C$ will be within $\epsilon/2$ of the true mean $\mathbf{E}[\mathbf{W}^2_C]$. (Technically, we must also note that since θ is much smaller than 1 we can do rejection sampling with very little slowdown.) Thus it remains to show that indeed $|\mathbf{E}[(\mathbf{W}_C)^2] - s| \leq \epsilon/2$.

Observe that $\mathbf{E}[(\mathbf{W}_C)^2] = \sum_{i=1}^k \pi^i \mathbf{E}[(\mathbf{W}_C)^2 \mid i \text{ is chosen}]$ and $s = \sum_{i=1}^k \pi^i \mathbf{E}[\mathbf{W}^2 \mid i \text{ is chosen}]$. Thus by convexity it is sufficient to prove $|\mathbf{E}[(\mathbf{X}_1^i)^2(\mathbf{X}_2^i)^2 \mid (\mathbf{X}_1^i)^2(\mathbf{X}_1^i)^2 \leq C] - \mathbf{E}[(\mathbf{X}_1^i)^2(\mathbf{X}_2^i)^2]| \leq \epsilon/2$ for all $i = 1 \ldots k$. For simplicity we now write $\mathbf{X}_j = \mathbf{X}_j^i$ for $j = 1, 2$. Recall that $\mathbf{X}_1$ and $\mathbf{X}_2$ are one-dimensional $(\mu_{\max}, \sigma_{\min}^2, \sigma_{\max}^2)$-bounded Gaussians.

Let $p(w)$ be the pdf for the random variable $(\mathbf{X}_1)^2(\mathbf{X}_2)^2$. Note that

$$\begin{aligned}
\left| \int_{|w|>C} wp(w)dw \right| &= \int_{x_1}\int_{x_2} \mathbf{1}_{\{x_1^2 x_2^2 \geq C\}} x_1^2 x_2^2 \mathbf{X}_1(x_1)\mathbf{X}_2(x_2)dx_1dx_2 \\
&\leq \int_{x_1}\int_{x_2} (\mathbf{1}_{\{|x_1|\geq C^{1/4}\}} + \mathbf{1}_{\{|x_2|\geq C^{1/4}\}}) x_1^2 x_2^2 \mathbf{X}_1(x_1)\mathbf{X}_2(x_2)dx_1dx_2 \\
&= \int_{x_2} x_2^2 \mathbf{X}_2(x_2)dx_2 \int_{|x_1|\geq M} x_1^2 \mathbf{X}_1(x_1)dx_1 \\
&\qquad + \int_{x_1} x_1^2 \mathbf{X}_1(x_1)dx_1 \int_{|x_2|\geq M} x_2^2 \mathbf{X}_2(x_2)dx_2 \\
&= \mathbf{E}[(\mathbf{X}_2)^2] \int_{|x_1|\geq M} x_1^2 \mathbf{X}_1(x_1)dx_1 \\
&\qquad + \mathbf{E}[(\mathbf{X}_1)^2] \int_{|x_2|\geq M} x_2^2 \mathbf{X}_2(x_2)dx_2 \\
&\leq 2L^2 \left(\int_{|x_1|\geq M} x_1^2 \mathbf{X}_1(x_1)dx_1 + \int_{|x_2|\geq M} x_2^2 \mathbf{X}_2(x_2)dx_2 \right) \\
&\leq 4\theta L^2, \qquad (13)
\end{aligned}$$

using the definitions of M and L.

Let $\eta = 1/(1 - \Pr[(\mathbf{X}_1)^2(\mathbf{X}_2)^2 > C]) - 1$, so $\eta \leq 3\theta$ using the same argument as in (12). Note that the pdf $p_C(w)$ for the random variable $(\mathbf{X}_1)^2(\mathbf{X}_2)^2$ conditioned on $|(\mathbf{X}_1)^2(\mathbf{X}_2)^2| \leq C$ is given by

$$p_C(w) = \begin{cases} (1+\eta)p(w) & \text{if } |w| \leq C, \\ 0 & \text{if } |w| > C. \end{cases}$$

Let $t = \mathbf{E}[(\mathbf{X}_1)^2(\mathbf{X}_2)^2]$; finally, we can show that $|\mathbf{E}[(\mathbf{X}_1)^2(\mathbf{X}_2)^2 \mid (\mathbf{X}_1)^2(\mathbf{X}_2)^2 \leq C] - t| \leq \epsilon/2$, as desired:

$$\begin{aligned}
|\mathbf{E}[(\mathbf{X}_1)^2(\mathbf{X}_2)^2 \mid (\mathbf{X}_1)^2(\mathbf{X}_2)^2 \leq C] - t| &= \left| \int_{\mathbf{R}} wp_C(w) - \int_{\mathbf{R}} wp(w) \right| \\
&= \left| (1+\eta) \int_{|w|\leq C} wp(w) - \int_{|w|\leq C} wp(w) - \int_{|w|>C} wp(w) \right| \\
&= \left| \eta \int_{|w|<C} wp(w) - \int_{|w|\geq C} wp(w) \right| \\
&\leq \eta t + \theta \leq (3\theta)\text{poly}(L) + \theta,
\end{aligned}$$

once more using the definition of M (note: $C \geq M$). Choosing $\theta = \text{poly}(\epsilon/L)$, we get that this is bounded by $\epsilon/2$; consequently $M = \text{poly}(L/\epsilon)$ and the sampling time is as claimed. ■

D.3 Processing the candidates

In this section we define a process (similar to the one for product distributions over $\{0,1\}^n$) that converts a single estimate for the π^i's, μ^i_j's and σ^i_j's of $\mathbf{Z}$ to a mixture of Gaussians that has bounded mass on every point in $[-M,M]^n$, as required by the ML procedure. It also guarantees that if the parametric estimates are accurate (close to the true values for the unknown $\mathbf{Z}$), then the process outputs a distribution with small KL divergence relative to $\mathbf{Z}$.

We begin by stating some basic facts about the KL divergence of two Gaussians, the first of which can be found in, e.g., [21]:

Fact 21 *Let $\mathbf{P},\mathbf{Q}$ each be a one-dimensional normal distribution with means and variances $\mu_{\mathbf{P}},\sigma_{\mathbf{P}}$ and $\mu_{\mathbf{Q}},\sigma_{\mathbf{Q}}$ respectively. Then we have*

$$\mathrm{KL}(\mathbf{P}||\mathbf{Q}) = \frac{1}{2}\ln\left(\frac{\sigma^2_{\mathbf{Q}}}{\sigma^2_{\mathbf{P}}}\right) + \frac{(\mu_{\mathbf{P}}-\mu_{\mathbf{Q}})^2+\sigma^2_{\mathbf{P}}-\sigma^2_{\mathbf{Q}}}{2\sigma^2_{\mathbf{Q}}}.$$

An easy consequence is:

Corollary 22 *Let $\mathbf{P},\mathbf{Q}$ be one-dimensional Gaussians as above and suppose that $|\mu_{\mathbf{P}}-\mu_{\mathbf{Q}}| \le \epsilon_{\mathrm{means}}$, $|\sigma^2_{\mathbf{P}}-\sigma^2_{\mathbf{Q}}| < \epsilon_{\mathrm{vars}}$, and $\sigma^2_{\mathbf{P}} \ge \sigma^2_{\min}$. Then*

$$\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \le \frac{\epsilon_{\mathrm{vars}}}{2\sigma^2_{\min}} + \frac{\epsilon^2_{\mathrm{means}}+\epsilon_{\mathrm{vars}}}{2(\sigma^2_{\min}-\epsilon_{\mathrm{vars}})}.$$

Proof: We have

$$\frac{\sigma^2_{\mathbf{Q}}}{\sigma^2_{\mathbf{P}}} \le \frac{\sigma^2_{\min}-\epsilon_{\mathrm{vars}}}{\sigma^2_{\min}} = 1+\frac{\epsilon_{\mathrm{vars}}}{\sigma^2_{\min}}$$

which implies

$$\frac{1}{2}\ln\left(\frac{\sigma^2_{\mathbf{Q}}}{\sigma^2_{\mathbf{P}}}\right) \le \frac{\epsilon_{\mathrm{vars}}}{2\sigma^2_{\min}}.$$

The bound easily follows observing that $\sigma^2_{\mathbf{Q}} \ge \sigma^2_{\min}-\epsilon_{\mathrm{vars}}$. ■

We now give the main theorem of the section:

Theorem 23 *There is a simple efficient procedure $\mathcal{A}$ which takes values $\hat{\pi}^i,\hat{\mu}^i_j,\hat{\sigma}^i_j$ and M as inputs and outputs a true mixture $\dot{\mathbf{Z}}$ of k $(\mu_{\max},\sigma^2_{\min},\sigma^2_{\max})$-bounded Gaussians with mixing weights $\dot{\pi}^1,\ldots,\dot{\pi}^k$ satisfying*

- $\sum_{i=1}^k \dot{\pi}^i = 1$, *and*
- $\alpha_{\mathrm{gauss}} \le \dot{\mathbf{Z}}(x) \le \beta_{\mathrm{gauss}}$ *for all* $x \in [-M,M]^n$,

where

$$\alpha_{\mathrm{gauss}} := \left[\frac{1}{\sqrt{2\pi}\sigma_{\max}}\cdot\exp\left(\frac{-2M^2}{\sigma^2_{\min}}\right)\right]^n, \qquad \beta_{\mathrm{gauss}} := 1/(\sqrt{2\pi}\sigma_{\min})^n.$$

Furthermore, suppose $\mathbf{Z}$ is a mixture of $(\mu_{\max},\sigma^2_{\min},\sigma^2_{\max})$-bounded Gaussians with mixing weights $\pi^1,\ldots,\pi^k$, means μ^i_j, and variances $(\sigma^i_j)^2$ and that the following are satisfied:

- *for $i=1\ldots k$ we have $|\pi^i-\hat{\pi}^i| \le \epsilon_{\mathrm{wts}}$, and*

- *for all i, j such that $\pi^i \geq \epsilon_{\text{minwt}}$ we have $|\mu_j^i - \hat{\mu}_j^i| \leq \epsilon_{\text{means}}$ and $|(\sigma_j^i)^2 - (\hat{\sigma}_j^i)^2| \leq \epsilon_{\text{vars}}$;*

then $\dot{\mathbf{Z}}$ will satisfy

$$\text{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq \eta_{\text{gauss}}(\epsilon_{\text{means}}, \epsilon_{\text{vars}}, \epsilon_{\text{wts}}, \epsilon_{\text{minwt}}),$$

where

$$\eta_{\text{gauss}} := n\left(\frac{\epsilon_{\text{vars}}}{2\sigma_{\min}^2} + \frac{\epsilon_{\text{means}}^2 + \epsilon_{\text{vars}}}{2(\sigma_{\min}^2 - \epsilon_{\text{vars}})}\right) + k\epsilon_{\text{minwt}} \cdot n\left[\frac{\sigma_{\max}^2 + 2\mu_{\max}^2}{\sigma_{\min}^2}\right] + \epsilon_{\text{wts}}^{1/3}.$$

Proof: We construct a mixture $\dot{\mathbf{Z}}$ of product distributions $\dot{\mathbf{Z}}^1, \ldots, \dot{\mathbf{Z}}^k$ by defining new mixing weights $\dot{\pi}^i$, expectations $\dot{\mu}_j^i$, and variances $(\dot{\sigma}_j^i)^2$. The procedure $\mathcal{A}$ is defined as follows:

1. For all i, j, set
$$\dot{\mu}_j^i = \begin{cases} -\mu_{\max} & \text{if } \hat{\mu}_j^i < -\mu_{\max} \\ \mu_{\max} & \text{if } \hat{\mu}_j^i > \mu_{\max} \\ \hat{\mu}_j^i & \text{o.w.} \end{cases}$$

2. For all i, j let
$$\dot{\sigma}_j^i = \begin{cases} \sigma_{\min} & \text{if } \dot{\sigma}_j^i < \sigma_{\min} \\ \sigma_{\max} & \text{if } \dot{\sigma}_j^i > \sigma_{\max} \\ \hat{\sigma}_j^i & \text{o.w.} \end{cases}$$

3. For all $i = 1, \ldots, k$ let
$$\ddot{\pi}^i = \begin{cases} \hat{\pi}^i & \text{if } \hat{\pi}^i \geq \epsilon_{\text{wts}} \\ \epsilon_{\text{wts}} & \text{if } \hat{\pi}^i < \epsilon_{\text{wts}} \end{cases}$$

   Let s be such that $s\sum_{i=1}^k \ddot{\pi}^i = 1$. Take $\dot{\pi}^i = s\ddot{\pi}^i$. (Note that this definition of $\dot{\pi}^i$ is identical to what was done in the proof of Theorem 19 for product distributions over $\{0,1\}^n$.)

Consider some particular μ_j^i and $\dot{\mu}_j^i$ and σ_j^i and $\dot{\sigma}_j^i$ where i is such that $\pi^i \geq \epsilon_{\text{minwt}}$, so we have $|\mu_j^i - \hat{\mu}_j^i| \leq \epsilon_{\text{means}}$ and $|(\sigma_j^i)^2 - (\hat{\sigma}_j^i)^2| \leq \epsilon_{\text{vars}}$. Since $|\mu_j^i| \leq \mu_{\max}$, by the definition of $\dot{\mu}_j^i$ we have that $|\mu_j^i - \dot{\mu}_j^i| \leq \epsilon_{\text{means}}$, and likewise that $|(\sigma_j^i)^2 - (\dot{\sigma}_j^i)^2| \leq \epsilon_{\text{vars}}$. Let $\mathbf{P}$ and $\mathbf{Q}$ be the one-dimensional Gaussians with means μ_j^i and $\dot{\mu}_j^i$ and variances σ_j^i and $\dot{\sigma}_j^i$ respectively. By Corollary 22, we have

$$\text{KL}(\mathbf{P}||\mathbf{Q}) \leq \frac{\epsilon_{\text{vars}}}{2\sigma_{\min}^2} + \frac{\epsilon_{\text{means}}^2 + \epsilon_{\text{vars}}}{2(\sigma_{\min}^2 - \epsilon_{\text{vars}})}.$$

Each $\dot{\mathbf{Z}}^i$ is the product of n such Gaussians. Therefore, by Proposition 17, we have for all i,

$$\text{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i) \leq n\left(\frac{\epsilon_{\text{vars}}}{2\sigma_{\min}^2} + \frac{\epsilon_{\text{means}}^2 + \epsilon_{\text{vars}}}{2(\sigma_{\min}^2 - \epsilon_{\text{vars}})}\right).$$

Recalling our bounds on $\dot{\sigma}_j^i$ and $\dot{\mu}_j^i$, we have

$$\dot{\mathbf{Z}}^i(x) \leq 1/(\sqrt{2\pi}\sigma_{\min})^n := \beta_{\text{gauss}}$$

for all $x \in [-M, M]$, and hence the same lower bound holds for $\dot{\mathbf{Z}}(x)$. Similarly, using the fact that $M \geq \mu_{\max}$ we have that

$$\dot{\mathbf{Z}}^i(x) \geq \left[\frac{1}{\sqrt{2\pi}\sigma_{\max}} \cdot \exp\left(\frac{-2M^2}{\sigma_{\min}^2}\right)\right]^n := \alpha_{\text{gauss}}$$

for all $x \in [-M, M]^n$ and similarly this upper bound also holds for $\dot{\mathbf{Z}}(x)$. We also have that for all i

$$\mathrm{KL}(\mathbf{Z}^i || \dot{\mathbf{Z}}^i) \leq n \left[\frac{\sigma_{\max}^2 + 2\mu_{\max}^2}{\sigma_{\min}^2} \right],$$

which follows from Fact 21 and Proposition 17.

Our goal is to apply Proposition 18 to bound $\mathrm{KL}(\dot{\mathbf{Z}}||\mathbf{Z})$. By the same argument as Theorem 19, we have that $|\pi^i - \dot{\pi}^i| \leq 3k\epsilon_{\mathrm{wts}}$ and $\dot{\pi}^i \geq \epsilon_{\mathrm{wts}}/2$ for all $i = 1, \ldots, k$. We apply Proposition 18 with the following parameters (note that the first four are exactly as in Theorem 19, only the value of ϵ is different):

$$\epsilon_1 = 3k\epsilon_{\mathrm{wts}}; \quad \epsilon_2 = \epsilon_{\mathrm{wts}}/2; \quad \epsilon_3 = \epsilon_{\mathrm{minwt}}; \quad \epsilon_4 = \epsilon_{\mathrm{wts}}^{1/2}; \quad \epsilon = n \left(\tfrac{\epsilon_{\mathrm{vars}}}{2\sigma_{\min}^2} + \tfrac{\epsilon_{\mathrm{means}}^2 + \epsilon_{\mathrm{vars}}}{2(\sigma_{\min}^2 - \epsilon_{\mathrm{vars}})} \right).$$

Proposition 18 now gives us:

$$\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq n \left(\frac{\epsilon_{\mathrm{vars}}}{2\sigma_{\min}^2} + \frac{\epsilon_{\mathrm{means}}^2 + \epsilon_{\mathrm{vars}}}{2(\sigma_{\min}^2 - \epsilon_{\mathrm{vars}})} \right) + k\epsilon_{\mathrm{minwt}} \cdot n \left[\frac{\sigma_{\max}^2 + 2\mu_{\max}^2}{\sigma_{\min}^2} \right] + \epsilon_{\mathrm{wts}}^{1/3}.$$

■

D.4 Proof of Theorem 4

Recall Theorem 4:

Theorem 4: *Let $\mathbf{Z}$ be any unknown mixture of k $(\mu_{\max}, \sigma_{\min}^2, \sigma_{\max}^2)$-bounded Gaussians. Let $M = M(\mathrm{poly}(1/n, 1/L, \epsilon))$. There is a $(Ln/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$ outputs a list of $(Ln/\epsilon)^{O(k^3)}$ many mixtures of $(\mu_{\max}, \sigma_{\min}^2, \sigma_{\max}^2)$-bounded Gaussians with the property that with probability $1 - \delta$,*

- *every distribution $\mathbf{Z}'$ in the list satisfies $\exp(-\mathrm{poly}(n, L, 1/\epsilon)) \leq \mathbf{Z}'(x) \leq \mathrm{poly}(L)^n$ for all $x \in [-M, M]^n$, and*
- *some distribution $\mathbf{Z}^\star$ in the list satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$.*

Proof: Let $\epsilon, \delta > 0$ be given. Run Mix-A-Lot with $\epsilon_{\mathrm{means}} = \frac{\epsilon\sigma_{\min}^3}{12n}$, $\epsilon_{\mathrm{vars}} = 2\epsilon_{\mathrm{means}}$, $\epsilon_{\mathrm{minwt}} = \frac{\epsilon\sigma_{\min}^2}{3kn(\sigma_{\max}^2 + 2\mu_{\max}^2)}$ and $\epsilon_{\mathrm{wts}} = \frac{\epsilon^3\sigma_{\min}^6}{120k^2n^3\mu_{\max}(\sigma_{\max}^2 + 2\mu_{\max}^2)^2}$. Note that $\epsilon_{\mathrm{wts}} < \epsilon_{\mathrm{means}}\epsilon_{\mathrm{minwt}}^{3/2}/\mu_{\max}$ as required by Theorem 1. This takes time $(Ln/\epsilon)^{O(k^3)} \cdot \log \frac{1}{\delta}$. We get as output a list of $(Ln/\epsilon)^{O(k^3)}$ many candidate parameter settings $\langle \{\hat{\pi}^i\}, \{\hat{\mu}_j^i\}, \{(\hat{\sigma}_j^i)^2\} \rangle$ with the guarantee that with probability $1 - \delta$ at least one of the settings satisfies

- for $i = 1 \ldots k$ we have $|\pi^i - \hat{\pi}^i| \leq \epsilon_{\mathrm{wts}}$, and
- for all i, j such that $\pi^i \geq \epsilon_{\mathrm{minwt}}$ we have $|\mu_j^i - \hat{\mu}_j^i| \leq \epsilon_{\mathrm{means}}$ and $|(\sigma_j^i)^2 - (\hat{\sigma}_j^i)^2| \leq \epsilon_{\mathrm{vars}}$.

We now pass all of these settings through Theorem 23 with M chosen as stated. Note that $M = \mathrm{poly}(n, L, 1/\epsilon)$. It follows that the resulting distributions satisfy $\exp(-\mathrm{poly}(n, L, 1/\epsilon)) \leq \mathbf{Z}'(x) \leq \mathrm{poly}(L)^n$ for all $x \in [-M, M]^n$, and one can check that under our setting of parameters we obtain $\eta_{\mathrm{gauss}} \leq \epsilon$, so one of the resulting distributions $\mathbf{Z}^\star$ satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$. ■

E Proof of Theorem 5: Maximum Likelihood

Before proving Theorem 5 we give some preliminaries. Let $\mathbf{P}$ and $\mathbf{Q}$ be arbitrary distributions over some space X. We can rewrite the KL divergence between $\mathbf{P}$ and $\mathbf{Q}$ as

$$\mathrm{KL}(\mathbf{P}||\mathbf{Q}) = -H(\mathbf{P}) - \int_{x\in X} \mathbf{P}(x)\log \mathbf{Q}(x), \tag{14}$$

where $H(\mathbf{P}) = -\int_{x\in X}\mathbf{P}(x)\log\mathbf{P}(x)$ is the entropy of $\mathbf{P}$.

Consider the random variable $-\log\mathbf{Q}(x)$, where x is a sample from the distribution $\mathbf{P}$. Using (14), we can express the expectation of this variable in terms of the KL-divergence:

$$\mathbf{E}_{x\in\mathbf{P}}[-\log\mathbf{Q}(x)] \quad = \quad \mathrm{KL}(\mathbf{P}||\mathbf{Q}) + H(\mathbf{P}). \tag{15}$$

Recall that when the ML algorithm runs on a list $\mathcal{Q}$ of distributions, it uses a set $\mathcal{S}$ of independent samples from $\mathbf{P}$, where $m = |\mathcal{S}|$. For each distribution $\mathbf{Q}\in\mathcal{Q}$, the algorithm computes

$$\Lambda(\mathbf{Q}) = \sum_{x\in\mathcal{S}}(-\log\mathbf{Q}(x)).$$

So, by (15), we have that the expected "score" of distribution $\mathbf{Q}$ is the following:

$$\mathbf{E}_{\mathcal{S}}[\Lambda(\mathbf{Q})] \quad = \quad m(H(\mathbf{P}) + \mathrm{KL}(\mathbf{P}||\mathbf{Q})). \tag{16}$$

We recall the theorem of Hoeffding [13]:

Theorem 24 *(Hoeffding) Let $x_1,\ldots,x_n$ be independent bounded random variables such that each x_i falls into the interval $[a,b]$ with probability one. Let $X = \sum_{i=1}^n x_i$. Then for any $t>0$ we have*

$$\Pr[X - \mathbf{E}[X] \geq t] \leq e^{-2t^2/n(b-a)^2} \quad \textit{and} \quad \Pr[X - \mathbf{E}[X] \leq -t] \leq e^{-2t^2/n(b-a)^2}.$$

Now we can prove Theorem 5:

Theorem 5 *Let β, α, $\epsilon > 0$ be such that $\alpha < \beta$. Let $\mathcal{Q}$ be a set of hypothesis distributions for some distribution $\mathbf{P}$ over the space X such that at least one $\mathbf{Q}^*\in\mathcal{Q}$ has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^*)\leq\epsilon$. Suppose also that $\alpha\leq\mathbf{Q}(x)\leq\beta$ for all $\mathbf{Q}\in\mathcal{Q}$ and all x such that $\mathbf{P}(x)>0$.*

Run the ML-algorithm *on $\mathcal{Q}$ using a set $\mathcal{S}$ of independent samples from $\mathbf{P}$, where $\mathcal{S}=m$. Then, with probability $1-\delta$, where*

$$\delta \quad \leq \quad (|\mathcal{Q}|+1)\cdot\exp\left(-2m\frac{\epsilon^2}{\log^2(\beta/\alpha)}\right),$$

the algorithm outputs some distribution $\mathbf{Q}^{\mathrm{ML}}\in\mathcal{Q}$ which has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^{\mathrm{ML}})\leq 4\epsilon$.

Proof: Call a distribution $\mathbf{Q}\in\mathcal{Q}$ *good* if $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^{\mathrm{ML}})\leq 4\epsilon$, and *bad* otherwise. Note that by assumption, we have at least one good distribution in $\mathcal{Q}$.

The probability δ that the algorithm fails to output some good distribution is at most the probability that either some bad distribution $\mathbf{Q}$ has $\Lambda(\mathbf{Q})\leq m(H(\mathbf{P})+3\epsilon)$ or the good distribution $\mathbf{Q}^*$ has $\Lambda(\mathbf{Q}^*)\geq m(H(\mathbf{P})+2\epsilon)$. Thus, by a union bound, we have

$$\delta \;\leq\; |\mathcal{Q}|\cdot\Pr\big[\Lambda(\mathbf{Q})\leq m(H(\mathbf{P})+3\epsilon)\mid \mathrm{KL}(\mathbf{P}||\mathbf{Q})\geq 4\epsilon\big] \;+\; \Pr\big[\Lambda(\mathbf{Q}^*)\geq m(H(\mathbf{P})+2\epsilon)\big] \tag{17}$$

For each bad $\mathbf{Q} \in \mathcal{Q}$ which has $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) > 4\epsilon$, we have

$$\begin{aligned}
\Pr[\Lambda(\mathbf{Q}) \leq m(H(\mathbf{P}) + 3\epsilon)] &= \Pr[\Lambda(\mathbf{Q}) \leq m(H(\mathbf{P}) + 4\epsilon) - \epsilon m)] \\
&\leq \Pr[\Lambda(\mathbf{Q}) \leq m(H(\mathbf{P}) + \mathrm{KL}(\mathbf{P}||\mathbf{Q})) - \epsilon m)] \quad (18) \\
&= \Pr[\Lambda(\mathbf{Q}) \leq \mathbf{E}_{\mathcal{S}}[\Lambda(\mathbf{Q})] - \epsilon m] \quad (19) \\
&\leq \exp\left(-2m\frac{\epsilon^2}{\log^2(\beta/\alpha)}\right). \quad (20)
\end{aligned}$$

Equation (18) follows from the bound on the KL-divergence, equation (19) follows from (16), and equation (20) follows from the Hoeffding bound (Theorem 24).

Following the same logic for $\mathbf{Q}^*$ where $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^*) \leq \epsilon$, we get

$$\begin{aligned}
\Pr[\Lambda(\mathbf{Q}^*) \geq m(H(\mathbf{P}) + 2\epsilon)] &= \Pr[\Lambda(\mathbf{Q}^*) \geq m(H(\mathbf{P}) + \epsilon) + m\epsilon] \\
&\leq \Pr[\Lambda(\mathbf{Q}^*) \geq m(H(\mathbf{P}) + \mathrm{KL}(\mathbf{P}||\mathbf{Q}^*)) + m\epsilon] \\
&= \Pr[\Lambda(\mathbf{Q}^*) \geq \mathbf{E}_{\mathcal{S}}[\Lambda(\mathbf{Q}^*)] + m\epsilon] \\
&\leq \exp\left(-2m\frac{\epsilon^2}{\log^2(\beta/\alpha)}\right). \quad (21)
\end{aligned}$$

The theorem follows from plugging equations (20) and (21) into equation (17). ■

F Truncated versus untruncated mixtures of Gaussians

Definition 4 *Let* $\mathbf{X}$ *be a distribution over* $\mathbf{R}^n$. *The* M-truncated version of $\mathbf{X}$ *is the distribution* $\mathbf{X}_M$ *obtained by restricting the support of* $\mathbf{X}$ *to be* $[-M, M]^n$ *and scaling so that* $\mathbf{X}_M$ *is a distribution. More precisely, for* $x \in \mathbf{R}^n$ *we have*

$$\mathbf{X}_M(x) = \begin{cases} 0 & \text{if } \|x\|_\infty > M, \\ c\mathbf{X}(x) & \text{if } \|x\|_\infty \leq M \end{cases}$$

where $c = 1/\left(\int_{x\in[-M,M]^n} \mathbf{X}(x)\right)$ *is chosen so that* $\int \mathbf{X}_M(x) = 1$.

In this section we prove Proposition 8:

Proposition 8 *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *be any mixtures of* n*-dimensional* $(\mu_{\max}, \sigma^2_{\min}, \sigma^2_{\max})$*-bounded Gaussians. Let* $\mathbf{P}_M$ *denote the* M*-truncated version of* $\mathbf{P}$, *where* M *is chosen as in Theorem 4. Then we have* $|\mathrm{KL}(\mathbf{P}_M||\mathbf{Q}) - \mathrm{KL}(\mathbf{P}||\mathbf{Q})| \leq 4\epsilon + 2\epsilon \cdot \mathrm{KL}(\mathbf{P}||\mathbf{Q})$.

Proof: We have that $\mathbf{P}_M(x)$ satisfies

$$\mathbf{P}_M(x) = \begin{cases} (1+\delta)\mathbf{P}(x) & \text{if } x \in [-M, M]^n, \\ 0 & \text{if } x \notin [-M, M]^n, \end{cases}$$

where $\delta > 0$ is chosen so that $\frac{1}{1+\delta} = \int_{x\in[-M,M]^n} \mathbf{P}(x)$. Using the definition of M we have

$$\int_{x\notin[-M,M]^n} \mathbf{P}(x) = \Pr_{\mathbf{P}}[x \notin [-M, M]^n] \leq \sum_{j=1}^{n} \Pr_{\mathbf{P}}[|x_j| \geq M] \leq n\theta \leq \epsilon$$

where we have used the fact that $\theta \leq \epsilon/n$. Consequently we have $\frac{1}{1+\delta} \geq 1 - \epsilon$, so $\delta \leq 2\epsilon$.

We have

$$\begin{aligned}
&|\mathrm{KL}(\mathbf{P}_M||\mathbf{Q}) - \mathrm{KL}(\mathbf{P}||\mathbf{Q})| \\
&= \left|\int_{x\in[-M,M]^n} (1+\delta)\mathbf{P}(x)\ln\frac{(1+\delta)\mathbf{P}(x)}{\mathbf{Q}(x)} - \int_{x\in\mathbf{R}^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}\right| \\
&= \left|(1+\delta)\ln(1+\delta)\int_{x\in[-M,M]^n}\mathbf{P}(x) + \delta\int_{x\in[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)} - \int_{x\notin[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}\right| \\
&\leq (1+\delta)\ln(1+\delta) + \delta\left|\int_{x\in[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}\right| + \left|\int_{x\notin[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}\right| \\
&= \delta(1+\delta) + \delta|R| + |S|,
\end{aligned}$$

where $R := \int_{x\in[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}$ and $S := \int_{x\notin[-M,M]^n}\mathbf{P}(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}$. For succinctness let κ denote $\mathrm{KL}(\mathbf{P}||\mathbf{Q})$. Note that we have $\kappa = R + S$.

Suppose we show that $|S| \leq \epsilon$. Then since $\kappa = R + S$, we must have $|R| \leq \kappa + \epsilon$, and hence $|\mathrm{KL}(\mathbf{P}_M||\mathbf{Q}) - \kappa| \leq \delta(1+\delta) + \delta(\kappa+\epsilon) + \epsilon \leq 4\epsilon + 2\epsilon\kappa$ (using $\delta \leq 2\epsilon$), as desired. Thus we can complete the proof by showing $|S| \leq \epsilon$.

Let us analyze the integrand of S. Decompose $\mathbf{P}$ into its mixture components, i.e. $\mathbf{P}(x) = \sum_{i=1}^k \pi^i\mathbf{P}^i(x)$, where $\mathbf{P}^1,\ldots,\mathbf{P}^k$ are n-dimensional $(\mu_{\max},\sigma^2_{\min},\sigma^2_{\max})$-bounded Gaussians. Hence

$$S = \sum_{i=1}^k \pi^i \int_{x\notin[-M,M]^k}\mathbf{P}^i(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}.$$

We will show that for each i we have $|\int_{x\notin[-M,M]^k}\mathbf{P}^i(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}| \leq \epsilon$. It then follows that $|S| \leq \epsilon$ since $|S|$ is upper bounded by a convex combination of these quantities.

Let us now analyze the quantity $\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}$. We will show that for any $x \notin [-M,M]^k$, neither $\mathbf{P}(x)$ nor $\mathbf{Q}(x)$ can be either "too small" or "too large" as a function of $||x||_2^2$; hence $|\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}|$ will be of moderate size. We will prove this for $\mathbf{P}(x)$ using the fact that it is a mixture of n-dimensional $(\mu_{\max},\sigma^2_{\min},\sigma^2_{\max})$-bounded Gaussians; since this is also true of $\mathbf{Q}(x)$, the same bound will hold for it.

We will show that for all $i = 1,\ldots,k$ and all $x \in \mathbf{R}^n$ we have $\mathbf{P}^i(x) \in [t(x), T]$ where T is a quantity and $t(x)$ is a function that will both be defined below. Since $\mathbf{P}(x) = \sum_{i=1}^k \pi^i\mathbf{P}^i(x)$ is a convex combination of the $\mathbf{P}^i(x)$'s, the same bound will hold for $\mathbf{P}(x)$. Fix any i and consider the Gaussian $\mathbf{P}^i$. Since this Gaussian is axis-aligned, we have $\mathbf{P}^i(x) = \prod_{j=1}^n \phi_{\mu_j,\sigma_j^2}(x_j)$ for some pairs $(\mu_1,\sigma_1^2),\ldots,(\mu_n,\sigma_n^2)$ satisfying $|\mu_j| \leq \mu_{\max}$, $\sigma_j^2 \in [\sigma^2_{\min},\sigma^2_{\max}]$. (Here $\phi_{\mu,\sigma^2}(x)$ is the usual pdf $\phi_{\mu,\sigma^2}(x) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left(\frac{-(x-\mu)^2}{2\sigma^2}\right)$ for a one-dimensional Gaussian.) It is easy to see that for any x_j,

$$\frac{1}{\sqrt{2\pi}\sigma_{\max}}\exp\left(-\frac{x_j^2}{\sigma^2_{\min}} - \frac{\mu^2_{\max}}{\sigma^2_{\min}}\right) \leq \phi_{\mu_j,\sigma_j^2}(x_j) \leq \frac{1}{\sqrt{2\pi}\sigma_{\min}}.$$

Hence for all $x \in \mathbf{R}^n$ we have

$$t(x) := \left(\frac{\exp(-\mu^2_{\max}/\sigma^2_{\min})}{\sqrt{2\pi}\sigma_{\max}}\right)^n \exp\left(-\frac{||x||_2^2}{\sigma^2_{\min}}\right) \leq \mathbf{P}^i(x) \leq \left(\frac{1}{\sqrt{2\pi}\sigma_{\min}}\right)^n =: T \tag{22}$$

for all i, and so (22) holds true for $\mathbf{P}(x)$ as well. As stated earlier, the same argument also shows that (22) holds for $\mathbf{Q}(x)$. We conclude that for any x,

$$\begin{aligned}\left|\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}\right| &\leq |\ln t(x)| + |\ln T| \\ &= \left|-n\frac{\mu_{\max}^2}{\sigma_{\min}^2} - n\ln(\sqrt{2\pi}\sigma_{\max}) - \frac{||x||_2^2}{\sigma_{\min}^2}\right| + n\ln(1/\sqrt{2\pi}\sigma_{\min}) \\ &\leq O\left(n\frac{\mu_{\max}^2}{\sigma_{\min}^2}\ln\frac{\sigma_{\max}}{\sigma_{\min}}||x||_2^2\right).\end{aligned}$$

Recall that we want to show $|\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}| \leq \epsilon$. It clearly suffices to show that $\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)|\ln\frac{\mathbf{P}(x)}{\mathbf{Q}(x)}| \leq \epsilon$. By the above it suffices to show

$$O\left(n\frac{\mu_{\max}^2}{\sigma_{\min}^2}\ln\frac{\sigma_{\max}}{\sigma_{\min}}\right)\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)||x||_2^2 \leq \epsilon.$$

We have

$$\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)||x||_2^2 = \sum_{j=1}^n\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)x_j^2 \tag{23}$$

Fix j; we now bound $\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)x_j^2$. Recall that $\mathbf{P}^i(x) = \mathbf{P}_1^i(x_1)\cdots\mathbf{P}_n^i(x_n)$. We have

$$\begin{aligned}\int_{x\notin[-M,M]^n}\mathbf{P}^i(x)x_j^2 &\leq \sum_{\ell=1}^n\int_{x\in\mathbf{R}^n:|x_\ell|>M}\mathbf{P}^i(x)x_j^2 \\ &= \int_{x\in\mathbf{R}^n:|x_j|>M}\mathbf{P}^i(x)x_j^2 + \sum_{\ell\neq j}\int_{x\in\mathbf{R}^n:|x_\ell|>M}\mathbf{P}^i(x)x_j^2\end{aligned}$$

For the first integral of (24) above we have

$$\begin{aligned}\int_{x\in\mathbf{R}^n:|x_j|>M}\mathbf{P}^i(x)x_j^2 = \left(\prod_{\ell\neq j}\left[\int_{x_\ell\in\mathbf{R}}\mathbf{P}_\ell^i(x_\ell)dx_\ell\right]\right)\cdot\int_{|x_j|>M}\mathbf{P}_j^i(x_j)x_j^2dx_j &= \int_{|x_j|>M}\mathbf{P}_j^i(x_j)x_j^2dx_j \\ &\leq \theta \qquad (24)\end{aligned}$$

where the inequality is by the definition of M. For the second term of (24) above we have

$$\sum_{\ell\neq j}\int_{x\in\mathbf{R}^n:|x_\ell|>M}\mathbf{P}^i(x)x_j^2 = \sum_{\ell\neq j}\left[\left(\int_{|x_\ell|>M}\mathbf{P}_\ell^i(x_\ell)dx_\ell\right)\left(\int_{x_j\in\mathbf{R}}\mathbf{P}_j^i(x_j)x_j^2dx_j\right)\right] \tag{25}$$

where we have used the fact that for any ℓ' which is neither ℓ nor j we have

$$\int_{x_{\ell'}\in\mathbf{R}}\mathbf{P}_{\ell'}^i(x_{\ell'})dx_{\ell'} = 1.$$

Again the definition of M to bound the integral over variable x_ℓ in (25) above by θ, we have that (25) is at most

$$\begin{aligned}(n-1)\theta\int_{x_j\in\mathbf{R}}\mathbf{P}_j^i(x_j)x_j^2dx_j = (n-1)\theta\mathbf{E}_{\mathbf{P}_j^i}[x^2] &= (n-1)\theta\left(\mathrm{Var}_{\mathbf{P}_j^i}[x] + \mathbf{E}_{\mathbf{P}_j^i}[x]^2\right) \\ &= (n-1)\theta((\sigma_j^i)^2 + (\mu_j^i)^2) \\ &\leq (n-1)\theta(\sigma_{\max}^2 + \mu_{\max}^2) \qquad (26)\end{aligned}$$

where the inequality holds since $\mathbf{P}_j^i$ is a one-dimensional $(\mu_{\max}, \sigma_{\min}^2, \sigma_{\max}^2)$-bounded Gaussian.

Putting all the pieces together, we find that (23) is at most

$$n[\theta + (n-1)\theta(\sigma_{\max}^2 + \mu_{\max}^2)] \leq n^2\theta(\sigma_{\max}^2 + \mu_{\max}^2)$$

It follows that $|S| \leq n^2\theta(\sigma_{\max}^2 + \mu_{\max}^2) \cdot O(n^2 \frac{\mu_{\max}^2}{\sigma_{\min}^2} \ln \frac{\sigma_{\max}}{\sigma_{\min}})$; this is at most ϵ since we take θ to be polynomial in ϵ and $\sigma_{\min}^2$ and inverse polynomial in $n, \mu_{\max}^2, \sigma_{\max}^2$. ■

G Proof of Theorem 9

The following claim is used in the proof of Theorem 9:

Claim 25 *Let T be a k-leaf decision tree, let $b \in \{-1, 1\}$ be a bit, let $S = \{x \in \{0,1\}^n : T(x) = b\}$, and let $\mathcal{U}_S$ denote the uniform distribution over S. Then $\mathcal{U}_S$ is a mixture of k product distributions.*

Proof: We show that $\mathcal{U}_S$ is a mixture of ℓ product distributions, where ℓ is the number of leaves in T which are labeled with bit b. To see this, observe that the k leaves of T partition $\{0,1\}^n$ into k disjoint subsets, each consisting of those $x \in \{0,1\}^n$ which reach the corresponding leaf. For a leaf at depth d the corresponding subset is of size 2^{n-d} and consists of those $x \in \{0,1\}^n$ which satisfy the length-d conjunction defined by the path from the root to that leaf. Thus, choosing a uniform element of S can be performed by the following process: (i) choose a leaf whose label is b, where each leaf at depth d is chosen with probability proportional to $1/2^d$; and then (ii) choose a uniform random example from the set of examples which satisfy the conjunction corresponding to that leaf. The uniform distribution over examples which satisfy a given conjunction is easily seen to be a product distribution $\mathbf{X}$ over $\{0,1\}^n$ in which $\mathbf{E}[\mathbf{X}_i] \in \{0, \frac{1}{2}, 1\}$ for all $i = 1, \ldots, n$. It follows that the uniform distribution over S is a mixture of ℓ product distributions of this sort. ■

Theorem 9: *For any function $k(n)$, if there is a poly(n) time algorithm which learns a mixture of $k(n)$ product distributions over $\{0,1\}^n$, then there is a poly(n) time uniform distribution PAC learning algorithm which learns the class of all $k(n)$-leaf decision trees.*

Proof: We suppose that we are given access to an oracle $\mathrm{EX}(T, \mathcal{U})$ which, at each invocation, supplies a labeled example $(x, T(x)) \in \{0,1\}^n \times \{0,1\}$ where x is chosen from the uniform distribution $\mathcal{U}$ over $\{0,1\}^n$ and T is the unknown $k(n)$-leaf decision tree to be learned. We describe an efficient algorithm A' which with probability $1 - \delta$ outputs a hypothesis $h : \{0,1\}^n \to \{0,1\}$ which satisfies $\Pr_{\mathcal{U}}[h(x) \neq T(x)] \leq \epsilon$. The algorithm A' uses as a subroutine an algorithm A which learns a mixture of $k(n)$ product distributions. Let M be the number of examples required by algorithm A to learn an unknown mixture of $k(n)$ product distributions to accuracy $1 - \frac{\epsilon}{2}$ and confidence $1 - \frac{\delta}{3}$.

Algorithm A' works as follows:

1. Determine $b \in \{-1, 1\}$ such that with probability $1 - \frac{\delta}{3}$ tree T outputs b on at least $1/3$ of the inputs in $\{0,1\}^n$. Let S denote $\{x \in \{0,1\}^n : T(x) = b\}$, and let $\mathcal{U}_S$ denote the uniform distribution over S.

2. Run algorithm A using samples from the uniform distribution $\mathcal{U}_S$; simulate $\mathcal{U}_S$ by invoking $\mathrm{EX}(T, \mathcal{U})$, and using the only examples with labels $T(x) = b$. To be confident that algorithm A receives at least M examples from $\mathcal{U}_S$, we draw $\Theta(M \log(1/\delta))$ examples from $\mathrm{EX}(T, \mathcal{U})$. Let $\mathcal{D}'$ be the hypothesis which is the output of A.

3. Output the hypothesis $h : \{0,1\}^n \to \{-1,1\}$ which is defined as follows: given x, if $\mathcal{D}'(x) \leq \frac{1}{2 \cdot 2^n}$ then $h(x) = -b$ else $h(x) = b$.

We now verify the algorithm's correctness. Note first that Step 1 can easily be performed by making $O(\log \frac{1}{\delta})$ draws from $\mathrm{EX}(T,\mathcal{U})$ to obtain an empirical estimate of $\Pr_{\mathcal{U}}[T(x) = b]$. Assuming that $|S|$ is indeed at least $2^n/3$, a simple Chernoff bound shows that $O(M \log \frac{1}{\delta})$ draws from $\mathrm{EX}(T,\mathcal{U})$ suffice to obtain M examples with label b in Step 2 with probability $1 - \frac{\delta}{3}$. We run A on examples generated by $\mathcal{U}_S$, which by Claim 25 is a mixture of k product distributions. Consequently, with overall probability at least $1 - \delta$ the hypothesis $\mathcal{D}'$ generated in Step 2 satisfies $\|D' - \mathcal{U}_S\|_1 \leq \frac{\epsilon}{2}$.

Now observe that the hypothesis h in Step 3 disagrees with T on precisely those x which either (i) belong to S but have $\mathcal{D}'(x) < \frac{1}{2 \cdot 2^n}$; or (ii) do not belong to S but have $\mathcal{D}'(x) \geq \frac{1}{2 \cdot 2^n}$. Each x of type (i) contributes at least $\frac{1}{2 \cdot 2^n}$ toward $\|\mathcal{D}' - \mathcal{U}_S\|_1$ since $\mathcal{U}_S(x) \geq \frac{1}{2^n}$ for each $x \in S$. Each x of type (ii) also incurs at least $\frac{1}{2 \cdot 2^n}$ toward $\|\mathcal{D}' - \mathcal{U}_S\|_1$. Consequently, since $\|\mathcal{D}' - \mathcal{U}_S\|_1 \leq \frac{\epsilon}{2}$, there are at most $\epsilon 2^n$ points $x \in \{0,1\}^n$ on which h is wrong. Thus, we have shown that with probability at least $1 - \delta$, the hypothesis h is an ϵ-accurate hypothesis for T with respect to the uniform distribution as desired. ■

Remark 1: We note that our reduction to decision tree learning in fact only uses quite restricted mixtures of product distributions in which (i) the mixture coefficients are proportional to powers of 2, (ii) the supports of the product distributions in the mixture are mutually disjoint, and (iii) each product distribution is a uniform distribution over some subcube of $\{0,1\}^n$ (equivalently, each product distribution has each $\mathbf{E}[\mathbf{X}_i] \in \{-1, 0, 1\}$). Thus, even this restricted class of mixtures of $k(n)$ product distributions is as hard to learn as $k(n)$-leaf decision trees.

Remark 2: Known results of Blum *et al.* [4] imply that the class of $k(n)$-leaf decision trees unconditionally cannot be learned under the uniform distribution in time less than $n^{\log k(n)}$ in the model of learning from *statistical queries*.

A "Statistical Query" learning algorithm is only allowed to obtain statistical estimates (accurate to within some specified error tolerance) of properties of the distribution over pairs $(x, T(x))$, and does not have access to actual labeled examples $(x, T(x))$. The algorithm is "charged" more time for estimates with a higher precision guarantee; this is motivated by the fact that such high-precision estimates would normally be obtained, given access to random examples, by drawing a large sample and making an empirical estimate. (See [14] for a detailed description of the Statistical Query model.)

Note that our algorithm for learning mixtures of product distributions interacts with the data solely by constructing empirical estimates of probabilities; thus, when this algorithm is used in the reduction of Theorem 9, the resulting algorithm for learning decision trees is a Statistical Query algorithm. Thus the results of Blum *et al.* unconditionally imply that no algorithm with the same basic approach as our algorithm can learn mixtures of $k(n)$ product distributions in time less than $n^{\log k(n)}$.