Learning mixtures of product distributions over discrete domains

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

Ryan O'Donnell†
Microsoft Research
Redmond, WA
odonnell@microsoft.com

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

Abstract

We consider the problem of learning *mixtures of product distributions over discrete domains* in the distribution learning framework introduced by Kearns et al. [18]. We give a poly(n/ϵ) time algorithm for learning a mixture of k arbitrary product distributions over the n-dimensional Boolean cube $\{0,1\}^n$ to accuracy ϵ, for any constant k. Previous polynomial time algorithms could only achieve this for $k = 2$ product distributions; our result answers an open question stated independently in [8] and [14]. We further give evidence that no polynomial time algorithm can succeed when k is superconstant, by reduction from a notorious open problem in PAC learning. Finally, we generalize our poly(n/ϵ) time algorithm to learn any mixture of $k = O(1)$ product distributions over $\{0,1,\ldots,b\}^n$, for any $b = O(1)$.

*Supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship

†Some of this work was done while at the Institute for Advanced Study, 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.

‡Supported in part by NSF CAREER award CCF-0347282.

1 Introduction

1.1 Framework and motivation. In this paper we study *mixture distributions*. Given distributions $\mathbf{X}^1, \dots, \mathbf{X}^k$ over $\mathbf{R}^n$ and mixing weights $\pi^1, \dots, \pi^k$ that sum to 1, a draw from the mixture distribution $\mathbf{Z}$ is obtained by first selecting i with probability π^i and then making a draw from $\mathbf{X}^i$. Mixture distributions arise in many practical scientific situations as diverse as medicine, geology, and artificial intelligence; indeed, there are several textbooks devoted to the subject [24, 19].

Assuming that data arises as a mixture of some distributions from a class of distributions $\mathcal{C}$, it is natural to try to *learn* the parameters of the mixture components. Our work addresses the learning problem in the PAC-style model introduced by Kearns et al. [18]. In this framework we are given a class $\mathcal{C}$ of probability distributions over $\mathbf{R}^n$ and access to random data sampled from an unknown mixture $\mathbf{Z}$ of k unknown distributions from $\mathcal{C}$. The goal is to output a *hypothesis* mixture $\mathbf{Z}'$ of k distributions from $\mathcal{C}$ which (with high confidence), is ϵ-close to the unknown mixture. The learning algorithm should run in time $\mathrm{poly}(n/\epsilon)$. The standard notion of "closeness" between distributions $\mathbf{Z}$ and $\mathbf{Z}'$, proposed by Kearns et al. and used in this work, is the *Kullback-Leibler (KL) divergence* (or *relative entropy*), defined as $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') := \int_x \mathbf{Z}(x)\ln(\mathbf{Z}(x)/\mathbf{Z}'(x))$.[1]

In this paper we learn mixtures of *product distributions* over the Boolean cube $\{0,1\}^n$, and more generally over the b-ary cube $\{0,\dots,b-1\}^n$; i.e., the classes $\mathcal{C}$ will consist of distributions $\mathbf{X}^i$ whose n coordinates are mutually independent distributions over $\{0,1\}$ and $\{0,\dots,b-1\}$, respectively.[2] Such learning problems have been well studied in the past, as we now describe.

1.2 Related work. In [18] Kearns et al. gave efficient algorithms for learning mixtures of *Hamming balls*; these are product distributions over $\{0,1\}^n$ in which all the coordinate means $\mathbf{E}[\mathbf{X}^i_j]$ must be either p or $1-p$ for some unknown p which is fixed over all mixture components. Although these algorithms can handle mixtures with $k = O(1)$ many components, the fact that the components are Hamming balls rather than general product distributions is a very strong restriction. (The algorithms also have some additional restrictions: p has to be bounded away from $1/2$, and a more generous learning scenario is assumed in which the learner is in addition given oracle access to the target distribution $\mathbf{Z}$ — i.e. she can submit an input x and get back the probability mass $\mathbf{Z}$ assigns to x.)

More recently, Freund and Mansour [14] gave an efficient algorithm for learning a mixture of two general product distributions over $\{0,1\}^n$. Around the same time Cryan et al. [9, 8] gave an efficient algorithm for learning phylogenetic trees in the two-state general Markov model; for the special case in which the tree topology is a star, this gives an algorithm for learning an arbitrary mixture of two product distributions over $\{0,1\}^n$. Both [14] and [8] stated as an open question the problem of obtaining a polynomial-time algorithm for learning a mixture of $k > 2$ product distributions. Indeed, recent work of Mossel and Roch [20] on learning phylogenetic trees argues that the rank-deficiency of transition matrices is a major source of difficulty, and this may indicate why $k = 2$ has historically been a barrier — a two-row matrix can be rank-deficient only if one row is a multiple of the other, whereas the general case of $k > 2$ is much more complex.

In other related work, there is a vast literature in statistics on the general problem of analyzing mixture data — see [19, 22, 24] for surveys. To a large degree this work centers on trying to find the exact best mixture model (in terms of likelihood) which explains a given data sample; this is computationally intractable in general. In contrast, our main goal (and the goal of [18, 14, 9, 8, 20])

[1] We remind the reader (see e.g. [7]) that $\|\mathbf{Z} - \mathbf{Z}'\|_1 \le (2\ln 2)\sqrt{\mathrm{KL}(\mathbf{Z}||\mathbf{Z}')}$ where $\|\cdot\|_1$ denotes total variation distance; hence if the KL divergence is small, then the total variation distance is also small.

[2] Of course, the algorithm works for product distributions over Σ^n for any alphabet Σ with $|\Sigma| = b$; i.e., the names of the characters in the alphabet do not matter.

is to obtain *efficient* algorithms that produce ϵ-close hypotheses.

We also note that there has also been recent interest in learning mixtures of n-dimensional Gaussians from the point of view of *clustering* [10, 11, 2, 25]. In this framework one is given samples from a a mixture of "well-separated" Gaussians, and the goal is to classify each point in the sample according to which Gaussian it came from. We discuss the relationship between our scenario and this recent literature on Gaussians in Section 6; here we emphasize that throughout this paper we make no "separation" assumptions (indeed, no assumptions at all) on the component product distributions in the mixture.

Finally, the problem of learning discrete mixture distributions may have applications to other areas of theoretical computer science, such as database privacy [23, 6] and quantum complexity [1].

1.3 Our results. In this paper we give an efficient algorithm for learning a mixture of $k = O(1)$ many product distributions over $\{0,1\}^n$. Our main theorem is the following:

Theorem 1 *Fix any $k = O(1)$, and let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0,1\}^n$. Then there is an algorithm that, given samples from $\mathbf{Z}$ and any $\epsilon, \delta > 0$ as inputs, runs in time $\mathrm{poly}(n/\epsilon) \cdot \log(1/\delta)$ and with probability $1 - \delta$ outputs a mixture $\mathbf{Z}'$ of k product distributions over $\{0,1\}^n$ satisfying $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') \leq \epsilon$.*

We emphasize that our algorithm requires none of the additional assumptions — such as minimum mixing weights or coordinate means bounded away from 0, 1/2, or 1 — that appear in some work on learning mixture distributions.

Our algorithm runs in time $(n/\epsilon)^{k^3}$, which is polynomial only if k is constant; however, this dependence may be unavoidable. In Theorem 7 we give a reduction from a notorious open question in computational learning theory (the problem of learning decision trees of superconstant size) to the problem of learning a mixture of any superconstant number of product distributions over $\{0,1\}^n$. This implies that solving the mixture learning problem for any $k = \omega(1)$ would require a major breakthrough in learning theory, and suggests that Theorem 1 may be essentially the best possible.

We also generalize our result to learn a mixture of product distributions over $\{0,\ldots,b-1\}^n$ for any constant b:

Theorem 2 *Fix any $k = O(1)$ and $b = O(1)$, and let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0,\ldots,b-1\}^n$. Then there is an algorithm that, given samples from $\mathbf{Z}$ and any $\epsilon, \delta > 0$ as inputs, runs in time $\mathrm{poly}(n/\epsilon) \cdot \log(1/\delta)$ and with probability $1 - \delta$ outputs a mixture $\mathbf{Z}'$ of k product distributions over $\{0,\ldots,b-1\}^n$ satisfying $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}') \leq \epsilon$.*

Taking $b = k$, this gives a polynomial time algorithm for learning k-state Markov Evolutionary Trees with a star topology. (Note that the main result of Cryan et al. [9, 8] is an algorithm for learning two-state METs with an arbitrary topology; hence our result is incomparable to theirs.)

2 Overview of our approach

2.1 The WAM **algorithm.** The cornerstone of our overall learning algorithms is an algorithm we call WAM (for WEIGHTS AND MEANS). WAM is a general algorithm taking as input a parameter $\epsilon > 0$ and having access to samples from an unknown mixture $\mathbf{Z}$ of k product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$. Here each $\mathbf{X}^i = (\mathbf{X}^i_1, \ldots, \mathbf{X}^i_n)$ is an $\mathbf{R}^n$-valued random vector with independent coordinates. The goal of WAM is to output accurate estimates for all of the *mixing weights* π^i and *coordinate means* $\mu^i_j := \mathbf{E}[\mathbf{X}^i_j]$. Note that a product distribution over $\{0,1\}^n$ is completely specified by its coordinate means.

More precisely, WAM outputs a *list* of $\mathrm{poly}(n/\epsilon)$ many *candidates* $(\langle\hat{\pi}^1, \ldots, \hat{\pi}^k\rangle, \langle\hat{\mu}_1^1, \hat{\mu}_2^1, \ldots, \hat{\mu}_n^k\rangle)$; each candidate may be viewed as a possible estimate for the correct mixing weights and coordinate means. We will show that with high probability at least one of the candidates output by WAM is *parametrically accurate*; roughly speaking this means that the candidate is a good estimate in the sense that in the sense that $|\hat{\pi}^i - \pi^i| \leq \epsilon$ for each i and that $|\hat{\mu}_j^i - \mu_j^i| \leq \epsilon$ for each i and j. However there is a slight twist: if a mixing weight π^i is very low then WAM may not receive any samples from $\mathbf{X}^i$, and thus it is not reasonable to require WAM to get an accurate estimate for $\mu_1^i, \ldots, \mu_n^i$. On the other hand, if π^i is so low then it is not very important to get an accurate estimate for $\mu_1^i, \ldots, \mu_n^i$ because $\mathbf{X}^i$ has only a tiny effect on $\mathbf{Z}$. We thus make the following formal definition:

Definition 1 *A candidate* $(\langle\hat{\pi}^1, \ldots, \hat{\pi}^k\rangle, \langle\hat{\mu}_1^1, \hat{\mu}_2^1, \ldots, \hat{\mu}_n^k\rangle)$ *is said to be* parametrically ϵ-accurate *if:*

1. $|\hat{\pi}^i - \pi^i| \leq \epsilon$ *for all* $1 \leq i \leq k$*;*
2. $|\hat{\mu}_j^i - \mu_j^i| \leq \epsilon$ *for all* $1 \leq i \leq k$ *and* $1 \leq j \leq n$ *such that* $\pi^i \geq \epsilon$.

The main technical theorem in this paper, Theorem 4, shows that so long as the $\mathbf{X}^i$'s take values in a bounded range, WAM will with high probability output at least one candidate that is parametrically accurate. The proof of this theorem uses tools from linear algebra (singular value theory) along with a very careful error analysis.

Remark 3 *As will be clear from the proof of Theorem 4,* WAM *will succeed even if the mixture distributions* $\mathbf{X}^i$ *are only pairwise independent, not fully independent. This may be of independent interest.*

2.2 From WAM **to PAC learning (binary case).** As we noted already, in the binary case a product distribution on $\{0,1\}^n$ is completely specified by its n coordinate means; thus a candidate can essentially be viewed as a hypothesis mixture of product distributions. (This is not precisely correct, as the candidate mixing weights may not precisely sum to 1 and the candidate means might be outside the range $[0,1]$ by as much as ϵ.) To complete the learning algorithm described in Theorem 1 we must give an efficient procedure that takes the list output by WAM and identifies a candidate distribution that is close to $\mathbf{Z}$ in KL divergence, as required by Theorem 1. We do this in two steps:

1. We first give an efficient procedure that converts a parametrically accurate candidate into a proper hypothesis distribution that is close to $\mathbf{Z}$ in KL divergence. We apply this procedure to each candidate in the list output by WAM, and thus obtain a list of mixtures (hypotheses), at least one of which is close to $\mathbf{Z}$ in KL divergence.
2. We then show that a maximum-likelihood procedure can take a list of hypotheses, at least one of which is good (close to $\mathbf{Z}$ in KL divergence), and identify a single hypothesis which is good.

2.3 Larger alphabets. In the larger alphabet setting, $\mathbf{Z}$ is a mixture of k product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\{0, \ldots, b-1\}^n$. Now each mixture component $\mathbf{X}^i$ is defined by bn parameters $p_{j,\ell}^i$ (with $j = 1, \ldots, n$ and $\ell = 0, \ldots, b-1$) where $p_{j,\ell}^i$ is the probability that a draw from $\mathbf{X}_j^i$ yields ℓ. The simple but useful observation that underlies our extension to $\{0, \ldots, b-1\}^n$ is the following: just as any distribution over $\{0,1\}$ is completely specified by its mean, any distribution $\mathbf{X}_j^i$ over $\{0, \ldots, b-1\}$ is completely specified by its first $b-1$ moments $\mathbf{E}[\mathbf{X}_j^i], \mathbf{E}[(\mathbf{X}_j^i)^2], \ldots, \mathbf{E}[(\mathbf{X}_j^i)^{b-1}]$. Our approach is thus to run WAM $b-1$ times; for $\ell = 1, \ldots, b-1$ the ℓth run will sample from the mixture distribution given by converting each sample $(z_1, \dots, z_n)$ to the sample $(z_1^\ell, \dots, z_n^\ell)$. We then carefully combine the lists output by the runs of WAM, and follow similar steps to (1) and (2) above to find a good hypothesis in the combined list.

2.4 Outline. Most of the main body of this paper, Section 3, is dedicated to explaining the ideas behind the WAM algorithm and its proof of correctness. (The detailed algorithm and proof appear in Appendices A through C.) We discuss the application of WAM to the b-ary case in Section 4, and in Section 5 we detail our reduction from a notorious open question in computational learning theory. We conclude in Section 6 with a discussion of applications and future work.

The two steps outlined in Section 2.2 are conceptually straightforward, but the details are quite technical, and are given in Appendices E through G. The pieces are all put together to prove Theorems 1 and 2 in Appendix H.

3 The WAM Algorithm

In this section we describe our main algorithm, WAM. We assume a general mixture setting: WAM has access to samples from $\mathbf{Z}$, a mixture of k product distributions $\mathbf{X}^1, \dots, \mathbf{X}^k$ with mixing weights $\pi^1, \dots, \pi^k$. Each $\mathbf{X}^i = (\mathbf{X}_1^i, \dots, \mathbf{X}_n^i)$ is an n-dimensional vector-valued random variable. We will further assume that all components' coordinates are bounded in the range $[-1, 1]$; i.e., $\mathbf{X}^i \in [-1, 1]^n$ with probability 1. We have chosen $[-1, 1]$ for mathematical convenience; by scaling and translating samples we can get a theorem about any interval such as $[0, 1]$ or $[0, (b-1)^{b-1}]$, with an appropriate scaling of ϵ. We write $\mu_j^i := \mathbf{E}[\mathbf{X}_j^i] \in [-1, 1]$ for the mean of the jth coordinate of $\mathbf{X}^i$.

Our main theorem is the following:

Theorem 4 *There is an algorithm* WAM *with the following property: for any $k = O(1)$ and any $\epsilon, \delta > 0$,* WAM *runs in time* $\mathrm{poly}(n/\epsilon) \cdot \log(1/\delta)$ *and outputs a list of* $\mathrm{poly}(n/\epsilon)$ *many candidates, at least one which (with probability at least $1 - \delta$) is parametrically ϵ-accurate.*

We give the full proof of correctness in Appendix C. The remainder of this section is devoted to explaining the main ideas behind the algorithm and its analysis.

3.1 Overview of WAM**.** There is of course a brute-force way to come up with a list of candidates $(\langle \hat{\pi}^1, \dots, \hat{\pi}^k \rangle, \langle \hat{\mu}_1^1, \hat{\mu}_2^1, \dots, \hat{\mu}_n^k \rangle)$, at least one of which is parametrically ϵ-accurate: simply "try all possible values" for the parameters up to additive accuracy ϵ. In other words, try all values $0, \epsilon, 2\epsilon, 3\epsilon, \dots, 1$ for the mixing weights and all values $-1, -1+\epsilon, \dots, 1-\epsilon, 1$ for the means. We call this approach "gridding". Unfortunately there are $\Theta(n)$ parameters in a candidate so this naive gridding strategy requires time (and produces a list of length) $(1/\epsilon)^{\Theta(n)}$, i.e. exponential in n, which is clearly unacceptable.

The basic idea behind WAM is as follows: given all pairwise correlations between the coordinates of $\mathbf{Z}$, it can be shown that there are a *constant* number of "key" parameters that suffice to determine all others. Hence in polynomial time we can empirically estimate all the correlations, try all possibilities for the constantly many key parameters, and then determine the remaining $\Theta(n)$ parameters.

The main challenge in implementing this idea is that it is not at all *a priori* clear that the error incurred from gridding the key parameters does not "blow up" when these are used to determine the remaining parameters. The heart of our analysis involves showing that it suffices to grid the key parameters to granularity $\mathrm{poly}(\epsilon/n)$ in order to get final error ϵ.

3.2 The algorithm, and intuition for the analysis. We will now go over the steps of the algorithm WAM and at the same time provide an "intuitive" discussion of the analysis. A concise description of the steps of WAM is given in Appendix A for the reader's convenience. Throughout this section we will assume for the sake of discussion that the steps we take incur no error; a sketch of the actual error analysis appears in Section 3.3.

The first step of WAM is to "grid" the values of the mixing weights $\{\pi^i\}$ to granularity $\epsilon_{\mathrm{wts}} := \epsilon^3$. Since there are only constantly many mixing weights, this costs just a multiplicative factor of $\mathrm{poly}(1/\epsilon)$ in the running time. The remainder of the algorithm "assumes" that the values currently being gridded for the mixing weights are the nearly-correct values of the mixing weights. In fact, for the purposes of this intuitive description of WAM, we will simply assume we have exactly correct values.

The next step is simple: Suppose some s of the k mixing weights we have are smaller than ϵ. By the definition of being "ϵ-parametrically accurate", we are not obliged to worry about coordinates with such small mixing weights; hence we will simply forget about these mixture components completely and treat k as $k-s$ in what follows. (We assign arbitrary values for the candidate means of the forgotten components.) We may henceforth assume that $\pi^i \geq \epsilon > 0$ for all i.

The next step of algorithm WAM is to use samples from $\mathbf{Z}$ to estimate the pairwise correlations between the coordinates of $\mathbf{Z}$. Specifically, for all pairs of coordinates $1 \leq j < j' \leq n$, the algorithm WAM empirically estimates

$$\mathrm{corr}(j, j') = \mathbf{E}[\mathbf{Z}_j \mathbf{Z}_{j'}].$$

The estimation will be done to within additive accuracy $\epsilon_{\mathrm{matrix}} = \mathrm{poly}(\epsilon/n)$; specifically, $\epsilon_{\mathrm{matrix}} := \tau^{k+1}$, where $\tau := \epsilon^2/n^2$. With high (i.e. $1-\delta$) confidence we will get good such estimates in time $\mathrm{poly}(n/\epsilon)$. Again, for the purposes of this intuitive description of WAM we will henceforth assume we have exactly correct values for each value $\mathrm{corr}(j, j')$. (As an aside, this is the only part of the algorithm that uses samples from $\mathbf{Z}$; as we will shortly see, this justifies Remark 3.)

Observe that since $\mathbf{X}^i_j$ and $\mathbf{X}^i_{j'}$ are (pairwise) independent we have

$$\mathrm{corr}(j, j') = \mathbf{E}[\mathbf{Z}_j \mathbf{Z}_{j'}] = \sum_{i=1}^{k} \pi^i \mathbf{E}[\mathbf{X}^i_j \mathbf{X}^i_{j'}] = \sum_{i=1}^{k} \pi^i \mathbf{E}[\mathbf{X}^i_j]\mathbf{E}[\mathbf{X}^i_{j'}] = \sum_{i=1}^{k} \pi^i \mu^i_j \mu^i_{j'}.$$

Let us define

$$\tilde{\mu}^i_j = \sqrt{\pi^i}\mu^i_j$$

and write $\tilde{\mu}_j = (\tilde{\mu}^1_j, \tilde{\mu}^2_j, \ldots, \tilde{\mu}^k_j) \in [-1, 1]^k$ for $1 \leq j \leq n$. We thus have

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

where $\cdot$ denotes the dot product in $\mathbf{R}^k$. The remaining task for WAM is to determine all the values μ^i_j. Since WAM already has values for each π^i and each $\pi^i \geq \epsilon > 0$, it suffices for WAM to determine all the values $\tilde{\mu}^i_j$ and then divide by $\sqrt{\pi^i}$.

At this point WAM has empirically estimated values for all the pairwise dot products $\tilde{\mu}_j \cdot \tilde{\mu}_{j'}$, $j \neq j'$, and as mentioned, for intuitive purposes we are assuming all of these estimates are exactly correct. Let M denote the $k \times n$ matrix whose (i, j) entry is the unknown $\tilde{\mu}^i_j$; i.e., the jth column of M is $\tilde{\mu}_j$. The statement that WAM has all the dot products $\tilde{\mu}_j \cdot \tilde{\mu}_{j'}$ for $j \neq j'$ is equivalent to saying that WAM has all the *off-diagonal* entries of the Gram matrix $M^\top M$. We are thus led to what is essentially the central problem WAM solves:

Central Task: *Given (estimates) for the off-diagonal entries of the $n \times n$ Gram matrix $M^\top M$, generate (estimates of) all possible candidates for the entries of the $k \times n$ matrix M.*

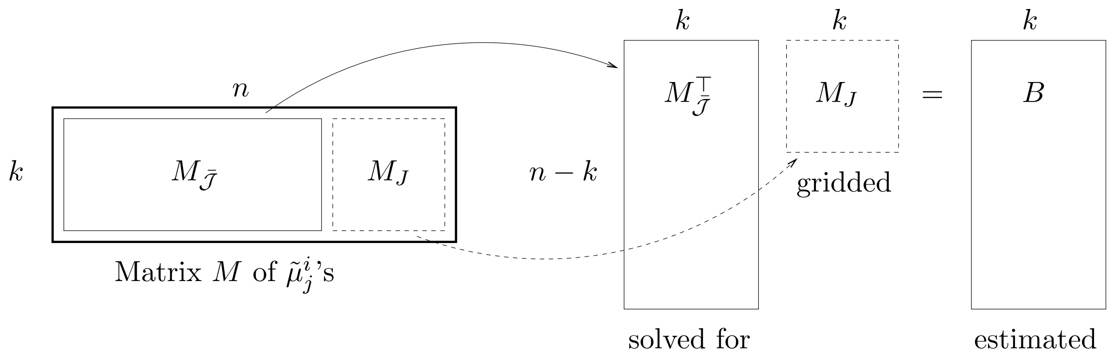

Figure 1: The full rank case. We solve for the unknown $\tilde{\mu}_j^i$'s in $M_{\bar{\mathcal{J}}}$.

(A remark: The diagonal entries of $M^\top M$ are the quantities $\tilde{\mu}_j \cdot \tilde{\mu}_j = \sum_{i=1}^k \pi^i (\mu_j^i)^2$ and there is no obvious way to estimate these quantities using samples from $\mathbf{Z}$. Also there are n such quantities, which is too many to "grid over". Nevertheless, the fact that we are missing the diagonal entries of $M^\top M$ will not play an important role for WAM.)

In general, a complete $n \times n$ Gram matrix determines the original $k \times n$ matrix matrix up to isometries on $\mathbf{R}^k$. Such isometries can be described by $k \times k$ orthonormal matrices, and these k^2 "degrees of freedom" roughly correspond to the constantly many key parameters that we grid over in the end. A geometric intuition for the Central Task is the following: there are n unknown vectors in $\mathbf{R}^k$ and we have all the "angles" between them (more precisely, the dot products) between them. Thus fixing k of the vectors (hence k^2 unknown coordinates) is enough to completely determine the remainder of the vectors.

The full rank case. We proceed with our intuitive description of WAM and show how to solve the Central Task *when M has full rank*. Having done this, we will give the actual steps of the algorithm that show how the full rank assumption can be removed.

So suppose for now that M has full rank. Then there exists some set of k columns of M that are linearly independent, say $J = \{j_1, \ldots, j_k\} \subset [n]$. Algorithm WAM tries all $\binom{n}{k} = \mathrm{poly}(n)$ possibilities for the set J and then grids over the vectors $\tilde{\mu}_{j_1}, \ldots, \tilde{\mu}_{j_k}$ with granularity $\epsilon_{\mathrm{matrix}} = \mathrm{poly}(\epsilon/n)$ in each coordinate. As usual for the purposes of intuition, we assume that we now have $\tilde{\mu}_{j_1}, \ldots, \tilde{\mu}_{j_k}$ 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. WAM now has the entries of M_J and must compute the remaining unknowns, $M_{\bar{\mathcal{J}}}$. Since WAM has all of 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 WAM can compute $M_{\bar{\mathcal{J}}}^\top = B M_J^{-1}$ in $\mathrm{poly}(n)$ time. Having done this, WAM has all the entries of M and so the Central Task is complete, as is the algorithm.

The general case. Of course in general, M does not have full rank. This represents the main conceptual problem we faced in rigorously solving the Central Task. Indeed, we believe that handling rank-deficiency is the chief conceptual problem for the whole learning mixtures question, and that our linear algebraic methods for overcoming it (the description of which occupies the remainder of Section 3) are the main technical contribution of this paper.

Suppose $\mathrm{rank}(M) = r < k$. By trying all possible values (only constantly many), algorithm WAM can be assumed to know r. Now by definition of $\mathrm{rank}(M) = r$ there must exist $k - r$

orthonormal vectors $u_{r+1}, \ldots, u_k \in [-1,1]^k$ which are orthogonal to all columns of M. WAM grids over these vectors with granularity ϵ_{matrix}, incurring another multiplicative poly(n/ϵ) time factor. As usual, assume for the intuitive discussion that we now have the u_j's exactly. Let these vectors be adjoined as columns to M, forming M'. But now the matrix M' has full rank; furthermore, WAM 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 the u_j's are simply 0! Thus we now have an instance of the Central Task with a full-rank matrix, a case we already solved. (Technically, n may now be as large as $n + (k-1)$, but this is still $O(n)$ and hence no time bounds are affected.) Given all entries of M' we certainly have all entries of M, and so we have solved the Central Task and completed the algorithm WAM in the rank-deficient case.

3.3 Sketch of the actual analysis of WAM**.** The preceding intuitive discussion of algorithm WAM neglected all error analysis. Correctly handling the error analysis is the somewhat subtle issue we discuss in this section. As mentioned, the full proof is given in Appendix C.

The main issue in the error analysis comes in understanding the right notion of the rank of M — since of all our gridding inevitably yields only approximations of the entries of M, the actual notion of rank is far too fragile to be of use. Recall the outline of the algorithm in our idealized intuition (rank-deficient case):

r = dimension of subspace in which $\tilde{\mu}_j$'s lie

$\Rightarrow$ augment M by $k - r$ orthogonal u_i's, forming $M' \Rightarrow M'$ now full rank

$\Rightarrow$ find nonsingular $k \times k$ submatrix $M'_{\mathcal{J}} \Rightarrow$ solve linear system ${M'}^\top_{\mathcal{J}} M'_{\mathcal{J}} = B$

For the purposes of the error analysis, we reinterpret the operation of WAM as follows:

r^* = dimension of subspace in which the $\tilde{\mu}_j$'s "essentially" lie

$\Rightarrow$ augment M by $k - r$ "essentially" orthogonal u_i's, forming $M' \Rightarrow M'$ now "strongly" full rank

$\Rightarrow$ find "strongly" nonsingular $k \times k$ submatrix $M'_{\mathcal{J}} \Rightarrow$ solve linear system ${M'}^\top_{\mathcal{J}} M'_{\mathcal{J}} = B$ (1)

The real difficulty of the error analysis comes in the last step: controlling the error incurred from the solution of the linear system. Since we will only have approximately correct values for the entries of $M'_{\mathcal{J}}$ and B, we need to analyze the additive error arising from solving a perturbed linear system. Standard results from numerical analysis (see Corollary 5 in Appendix B) let us bound this error by a function of: (i) the error in $M'_{\mathcal{J}}$ and B, and (ii) the smallest *singular value* of $M'_{\mathcal{J}}$, denoted by $\sigma_k(M')$.

Let us briefly recall some notions related to singular values: Given any $k \times n$ matrix M, the first (largest) singular value of M is $\sigma_1(M) = \max_{\|u_1\|_2=1} \|u_1^\top M\|_2$, and a u_1 achieving this maximum is taken as the first *(left) singular vector* of M. The second singular value of M is $\sigma_2(M) = \max_{\|u_2\|_2=1, u_2 \perp u_1} \|u_2^\top M\|_2$, and u_2 is the second left singular vector of M. In general, the ith singular value and vector are given by maximizing over all $\|u_i\|_2 = 1$ orthogonal to all $u_1, \ldots, u_{i-1}$. In a well-defined sense (the Frobenius norm), the smallest singular value $\sigma_k(M)$ measures the distance of M from being singular.

WAM's final error bounds arise from dividing the error in its estimates for $M'_{\mathcal{J}}$ and B by the smallest singular value of $M'_{\mathcal{J}}$. The error in the estimates for the entries of $M'_{\mathcal{J}}$ come from gridding, and thus can essentially be made as small as desired; WAM makes them smaller than ϵ_{matrix}. The errors in B come from two sources: some of the entries of B are estimates of quantities $\tilde{\mu}_j \cdot \tilde{\mu}_{j'} = \text{corr}(j, j')$, and again these errors can be made essentially as small as desired, smaller than ϵ_{matrix}. However the other errors in B come from approximating the quantities $\tilde{\mu}_j \cdot u_i$ by 0; i.e, assuming the augmenting vectors are orthogonal to the columns of M.

As the reader may by now have guessed, the vectors with which WAM attempts to augment M will be the last $k - r^*$ singular vectors of M, $u_{r^*+1}, \ldots, u_k$. The hope is that for an appropriate choice of r^*, these singular vectors will be "essentially" orthogonal to the columns of M, and that the resulting M' will be "strongly" full rank, in the sense that $\sigma_k(M')$ will be somewhat large (cf. (1)). One can show (see Proposition 9 of Appendix B) that the extent to which the u_i's are orthogonal to the columns of M is controlled by the (r^*+1)th singular value of M; i.e., $|\tilde{\mu}_j \cdot u_i| \leq \sigma_{r^*+1}(M)$ for all $i \geq r^* + 1$; this is precisely the error we incur for the zero entries in B. On the other hand, one can also show that the augmented M' has smallest singular value at least $\sigma_{r^*}(M)$. Thus we are motivated to choose r^* so as to get a large multiplicative gap between $\sigma_{r^*}(M)$ and $\sigma_{r^*+1}(M)$:

Definition 2 *Given $\tau > 0$, the* τ-essential rank *of M is*

$$r^*(M) = r^*_\tau(M) = \min\{0 \leq r \leq k : \sigma_{r+1}(M)/\sigma_r(M) \leq \tau\},$$

where we take $\sigma_0(M) = 1$ and $\sigma_{k+1}(M) = 0$.

One might think that if the additive error incurred from solving the linear system were to be roughly $\sigma_{r^*}(M)/\sigma_{r^*+1}(M)$ then it should suffice to select τ on the order of $\text{poly}(\epsilon)$. However, there is still a missing piece of the analysis: Although the smallest singular value of M' becomes at least $\sigma_{r^*}(M)$ after adjoining the u_j's, we only use a $k \times k$ *submatrix* $M'_{\mathcal{J}}$ to solve the linear system. Is it the case that if M' has a large smallest singular value then its "best" $k \times k$ submatrix also has a somewhat large smallest singular value? We need a quantitative version of the fact that a nonsingular $k \times n$ matrix has a $k \times k$ nonsingular submatrix (again, cf. (1)).

This does not seem to be a well-studied problem, and indeed there are some open questions in linear algebra surrounding the issue. It is possible to derive an extremely weak quantitative result of the required nature using the Cauchy-Binet formula. We instead give the following quantitatively strong version:

Corollary 5 *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 call the result a corollary because our proof in Appendix B is derived from a 1997 linear algebraic result of Goreinov, Tyrtyshnikov, and Zamarashkin [15]. Incidentally, it is conjectured in their paper, and we also conjecture, that $\sqrt{k(n-k)+1}$ can be replaced by $\sqrt{n}$.)

With this result in hand it becomes sufficient to take $\tau = \epsilon^2/n^2$, as described in the previous section. Now the error analysis can be completed:

- If M has a singular value gap of τ and so has essential rank $r^* < k$, then when WAM tries out the appropriate r^* and singular vectors, the error it incurs from solving the linear system is roughly at most $O(\sqrt{n}\tau) = O(\epsilon^2/n^{3/2})$; and as we show at the end of Appendix C, having this level of control over errors in solving the linear system for the unknown $\tilde{\mu}^i_j$'s lets us obtain the final μ^i_j values to the required ϵ-accuracy.

- On the other hand, if M has no singular value gap smaller than τ then its smallest singular value is at least τ^k to begin with; thus it suffices to take $\epsilon_{\text{matrix}} = \tau^{k+1} = \text{poly}(\epsilon/n)$ to control the errors in the full-rank case.

See Appendix C for the detailed proof of correctness.

4 Estimating Higher Moments

In this section we explain our remarks from Section 2.3 more thoroughly; specifically, how to use WAM to learn a mixture $\mathbf{Z}$ of k product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\{0, \ldots, b-1\}^n$. Such a distribution can be "parametrically" described by mixing weights $\{\pi^i\}_{i \in [k]}$ and probabilities $\{p^i_{j,\ell}\}$, where $p^i_{j,\ell} = \Pr[\mathbf{X}^i_j = \ell]$.

Running WAM on samples from $\mathbf{Z}$ gives a list of estimates of mixing weights and coordinate means $\mathbf{E}[\mathbf{X}^i_j]$, but these coordinate means are insufficient to completely describe the distributions $\mathbf{X}^i_j$. However, suppose that we run WAM on samples from $\mathbf{Z}^\ell$ (i.e. each time we obtain a draw $(z_1, \ldots, z_n)$ from $\mathbf{Z}$, we actually give $(z_1^\ell, \ldots, z_n^\ell)$ to WAM). It is easy to see that by doing this, we are running WAM on the π-weighted mixture of distributions $(\mathbf{X}^1)^\ell, \ldots, (\mathbf{X}^k)^\ell$; we will thus get as output a list of candidates for the mixing weights and the *coordinate ℓth moments* $\mathbf{E}[(\mathbf{X}^i_j)^\ell]$ for $\mathbf{Z}$.

Our algorithm for distributions over $\{0, \ldots, b-1\}^n$ uses this approach to obtain a list of candidate descriptions of each of the first $b-1$ coordinate moments of $\mathbf{Z}$. The algorithm then essentially takes the cross-product of these $b-1$ lists to obtain a list of overall candidates, each of which is an estimate of the mixing weights and all $b-1$ moments. Since WAM guarantees that each list contains an accurate estimate, the overall list will also contain an accurate estimate of the mixing weights and of all moments. For each candidate the estimate of the moments is then easily converted to "parametric form" $\{p^i_{j,\ell}\}$, and as we show, any candidate with accurate estimates of the moments yields an accurate estimate of the probabilities $p^i_{j,\ell}$.

We now give the main theorem of the section, the proof of which (in Appendix D) contains the details of the algorithm:

Theorem 6 *Fix $k = O(1), b = O(1)$. Let $\mathbf{Z}$ be a mixture of k product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\{0, \ldots, b-1\}^n$, so $\mathbf{Z}$ is described by mixing weights $\pi^1, \ldots, \pi^k$ and probabilities $\{p^i_{j,\ell}\}_{i \in [k], j \in [n], \ell \in \{0, \ldots, b-1\}}$.*

There is an algorithm with the following property: for any $\epsilon, \delta > 0$, the algorithm runs in poly$(n/\epsilon) \cdot \log \frac{1}{\delta}$ time and with probability $1-\delta$ outputs a list of candidates $\langle \{\hat{\pi}^i\}, \{\hat{p}^i_{j,\ell}\} \rangle$ such that for at least one candidate in the list, the following holds:

1. *$|\hat{\pi}^i - \pi^i| \leq \epsilon$ for all $i \in [k]$; and*
2. *$|\hat{p}^i_{j,\ell} - p^i_{j,\ell}| \leq \epsilon$ for all i, j, ℓ such that $\pi^i \geq \epsilon$.*

5 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.* 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 $b \in \{0, 1\}$. A decision tree T computes a Boolean function $f : \{0,1\}^n \to \{0,1\}$ 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:

Theorem 7 *For any function $k(n)$, if there is a poly(n/ϵ) time algorithm which learns a mixture of $k(n)$ many product distributions over $\{0,1\}^n$, then there is a poly(n/ϵ) time uniform distribution PAC learning algorithm which learns the class of all $k(n)$-leaf decision trees.*

The basic idea behind this theorem is quite simple. Given any $k(n)$-leaf decision tree T, the set of all positive examples for T is a union of at most $k(n)$ many disjoint subcubes of $\{0,1\}^n$, and thus the uniform distribution over the positive examples is a mixture of at most $k(n)$ product distributions over $\{0,1\}^n$. If we can obtain a high-accuracy hypothesis mixture $\mathcal{D}$ for this mixture of product distributions, then roughly speaking $\mathcal{D}$ must put "large" weight on the positive examples and "small" weight on the negative examples. We can thus use $\mathcal{D}$ to make accurate predictions of T's value on new examples very simply as follows: given a new example x to classify, we simply compute the probability weight that the hypothesis mixture $\mathcal{D}$ puts on x, and output 1 or 0 depending on whether this weight is large or small. We give the formal proof of Theorem 7 in Appendix I.

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.[3] The fastest algorithms to date [12, 3] can learn $k(n)$-leaf decision trees under the uniform distribution in time $n^{\log k(n)}$. This suggests that it may be impossible to learn mixtures of a superconstant number of product distributions over $\{0,1\}^n$ in polynomial time.

6 Conclusions and Future Work

We have shown how to learn mixtures of any constant number of product distributions over $\{0,1\}^n$, and more generally over $\{0,\dots,b-1\}^n$, in polynomial time.

The methods we use are quite general and can be adapted to learn mixtures of other types of multivariate product distributions which are definable in terms of their moments. Along these lines, we have used the approach in this paper to give a PAC-style algorithm for learning mixtures of $k = O(1)$ axis-aligned Gaussians in polynomial time [13]. (We note that while some previous work on learning mixtures of Gaussians from a clustering perspective can handle $k = \omega(1)$ many component Gaussians, all such work assumes that there is some minimum separation between the centers of the component Gaussians, since otherwise clustering is clearly impossible. In contrast, our result in [13] — in which we do not attempt to do clustering but instead find a hypothesis distribution with small KL-divergence from the target mixture — does not require us to assume that the component Gaussians are separated.) We expect that our techniques can also be adapted to learn mixtures of other distributions such as products of exponential distributions or beta distributions.

It is natural to ask if our approach can be extended to learn mixtures of distributions which are not necessarily product distributions; this is an interesting direction for future work. Note that our main algorithmic ingredient, algorithm WAM, only requires that that the coordinate distributions be pairwise independent.

Finally, one may also 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. Aaronson. Multilinear formulas and skepticism of quantum computation. In *Proceedings of the 36th Annual Symposium on Theory of Computing (STOC)*, pages 118–127, 2004.

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

[3] 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 [4].

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

[4] 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.

[5] 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.

[6] S. Chawla, C. Dwork, F. McSherry, A. Smith, and H. Wee. Towards privacy in public databases. To appear, Theory of Cryptography, 2005.

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

[8] M. Cryan. *Learning and approximation algorithms for problems motivated by evolutionary trees*. PhD thesis, University of Warwick, 1999.

[9] 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.

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

[11] 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.

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

[13] J. Feldman, R. O'Donnell, and R. Servedio. PAC Learning mixtures of axis-aligned Gaussians. manuscript, 2005.

[14] 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.

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

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

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

[18] 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.

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

[20] E. Mossel and S. Roch. Learning nonsingular phylogenies and hidden markov models. In *To appear in Proceedings of the 37th Annual Symposium on Theory of Computing (STOC)*, 2005.

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

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

[23] A. Smith. Personal communication. 2005.

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

[25] 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 WAM

Algorithm WAM has access to samples from the mixture $\mathbf{Z}$ and takes as input parameters $\epsilon, \delta > 0$.

`Algorithm` WAM:

1. Let $\epsilon_{\text{wts}} = \epsilon^3$, $\tau = \epsilon^2/n^2$, and $\epsilon_{\text{matrix}} = \tau^{k+1}$.

2. Grid over the mixing weights, producing values $\hat{\pi}^1, \ldots, \hat{\pi}^k \in [0,1]$ accurate to within $\pm\epsilon_{\text{wts}}$. If s of these weights are smaller than $\epsilon - \epsilon_{\text{wts}}$, eliminate them and treat k as $k - s$ in what follows.

3. 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 $\pm\epsilon_{\text{matrix}}$, with confidence $1 - \delta$.

4. Let M be the $k \times n$ matrix of unknowns $(M_{ij}) = (\tilde{\mu}^i_j)$, and try all possible integers $0 \leq r^* \leq k$ for the essential rank of M.

5. Grid over $k - r^*$ vectors $\hat{u}_{r^*+1}, \ldots, \hat{u}_k \in [-1,1]^k$ to within $\pm\epsilon_{\text{matrix}}$ in each coordinate and augment M with these as columns, forming $\widehat{M}'$.

6. Try all possible subsets 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. Grid over $[-1,1]$ for the entries of M in columns J to within $\pm\epsilon_{\text{matrix}}$, yielding $\{\hat{\tilde{\mu}}^i_j : i \in [k], j \in J\}$. Let $\widehat{M}'_{\mathcal{J}}$ denote the matrix of estimates for all the columns in $\mathcal{J}$. (See Figure 2.)

7. 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 $M^{\top}_{\bar{\mathcal{J}}} \widehat{M}'_{\mathcal{J}} = \hat{B}$ to obtain estimates $\hat{\tilde{\mu}}^i_j$ for the entries of $M_{\bar{\mathcal{J}}}$ (which are the unknown $\tilde{\mu}^i_j$'s), thus producing estimates $\hat{\tilde{\mu}}^i_j$ for all entries of M. (If the matrix $\widehat{M}'_{\mathcal{J}}$ is singular, simply abandon the current gridding.)

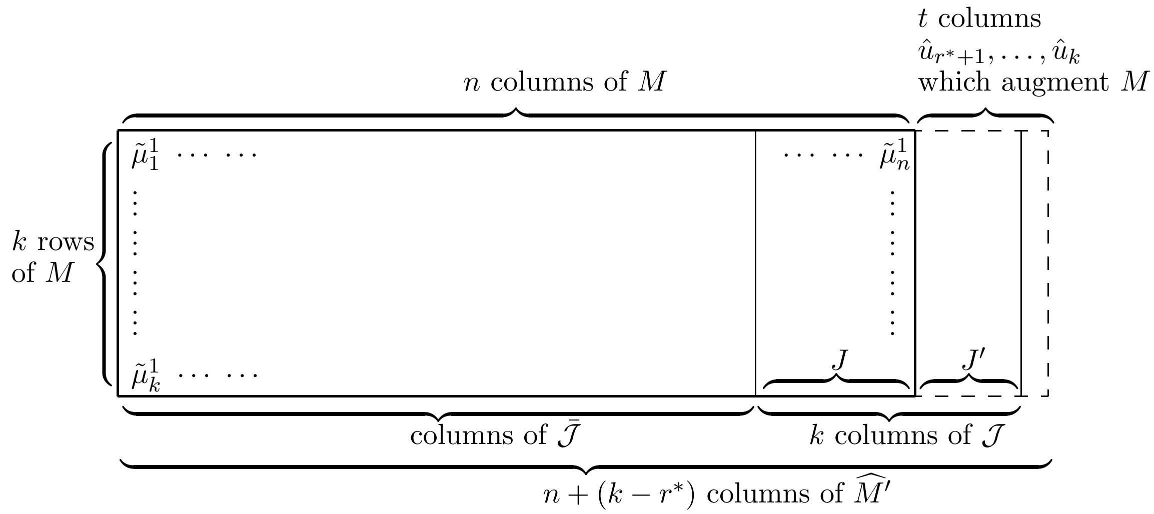

Figure 2: A depiction of the matrix used by WAM. For ease of illustration the columns J of M are depicted as being the rightmost columns of M, and the columns J' from the augmenting columns $\hat{u}_{k-t+1}, \ldots, \hat{u}_k$ are depicted as being the leftmost of those augmenting columns.

8. 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 each is at least $\epsilon - \epsilon_{\text{wts}} > 0$.)

9. Output the candidate $(\langle \hat{\pi}^1, \ldots, \hat{\pi}^k \rangle, \langle \hat{\mu}_1^1, \hat{\mu}_2^1, \ldots, \hat{\mu}_n^k \rangle)$.

B Linear algebra necessities

In this section we give the results from linear algebra and numerical analysis necessary for the analysis of WAM.

Let $A = (a_{ij})$ be any $k \times n$ real matrix and write its singular value decomposition as $A = U \Sigma V$. We let $\sigma_1 \geq \cdots \geq \sigma_k \geq 0$ denote the singular values of A, and let $u_1, \ldots, u_k$ denote the corresponding left singular vectors of A, i.e., the columns of U. 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 $\|A\|_2 = \max_{\|x\|_2 = 1} \|Ax\|$. It is well known that $\|A\|_2 = \sigma_1$ and $\|A\|_F = \sqrt{\sigma_1^2 + \cdots + \sigma_k^2}$; note that this implies $\|A\|_2 \leq \|A\|_F$.

Our first necessary result is a quantitative version of the elementary fact that a full-rank $k \times n$ matrix has a full-rank $k \times k$ submatrix. We will use the following theorem of Goreinov, Tyrtyshnikov, and Zamarashkin [15]:

Theorem 8 [15] *Let V be a $k \times n$ real matrix with orthonormal rows. Then there is a $k \times k$ submatrix V_J which has $\sigma_k(V_J) \geq 1/\sqrt{k(n-k)+1}$.*

The result we need is an easy corollary:

Corollary 5 *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 $\sigma_k(A_J) \geq \epsilon/\sqrt{k(n-k)+1}$.*

Proof: 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 8, so $\sigma_k(V_J) \geq 1/\sqrt{k(n-k)+1}$. We have $\sigma_k(U) = 1$ (since U is an orthogonal matrix) and $\sigma_k(\Sigma) \geq \epsilon$, so

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

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 $\sigma_k(P) = \min_{\|x\|_2=1} \|x^\top P\|_2$.) The corollary follows by observing that $U\Sigma V_J$ is the $k \times k$ submatrix of A whose columns are in J. ■

The next result we will need is the characterization of what happens when the last $k - r^*$ left singular vectors of a matrix are adjoined to it:

Proposition 9 *Let A be a $k \times n$ matrix with columns $a_1, \ldots, a_n$. Fix any r^* and let $u_{r^*+1}, \ldots, u_k$ be the left singular vectors corresponding to the smallest singular values $\sigma_{r^*+1}, \ldots, \sigma_k$ of A. Let A' be A with the vectors $u_{r^*+1}, \ldots, u_k$ adjoined as columns. Then*

$$\sigma_k(A') \geq \min\{1, \sigma_{r^*}(A)\},$$

and for all $r^* + 1 \leq \ell \leq k$ and for all columns a_j of A we have

$$|a_j \cdot u_\ell| \leq \sigma_{r^*+1}(A).$$

Proof: Write the singular value decomposition $A = U\Sigma V$ 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 $\sigma_1 \geq \cdots \geq \sigma_k \geq 0$ on the diagonal, and V is a $k \times n$ matrix with orthonormal rows. It follows that for any vector $x \in \mathbf{R}^k$ we have

$$\|x^\top A\|_2^2 = \sigma_1^2(x^\top u_1)^2 + \cdots + \sigma_k^2(x^\top u_k)^2.$$

Let R denote the $k \times (k - r^*)$ matrix whose columns are $u_{r^*+1}, \ldots, u_k$, so we have $A' = [A\ R]$. It is easily verified that the left singular vectors of R are simply $u_{r^*+1}, \ldots, u_k$, while the singular values of R are all 1. Consequently we have

$$\|x^\top R\|_2^2 = (x^\top u_{r^*+1})^2 + \cdots + (x^\top u_k)^2$$

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

Now recall the variational characterization of $\sigma_k(A')$, namely $\sigma_k(A') = \min_{\|x\|_2=1} \|x^\top A'\|_2$. Since $\|x^\top A'\|_2 = \sqrt{\|x^\top A\|_2^2 + \|x^\top R\|_2^2}$, we have

$$\sigma_k(A') = \min_{\|x\|_2=1} \sqrt{\sigma_1^2(x^\top u_1)^2 + \cdots + \sigma_k^2(x^\top u_k)^2 + (x^\top u_{r^*+1})^2 + \cdots + (x^\top u_k)^2}. \tag{2}$$

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 $\|x\|_2 = 1$. If we let $\alpha_x = (x^\top u_{r^*+1})^2 + \cdots + (x^\top u_k)^2$ then the quantity inside the square root of (2) is at least $\sigma_{r^*}^2(1-\alpha_x) + \alpha_x \geq \min\{\sigma_{r^*}^2, 1\}$. This proves the first inequality of the proposition.

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

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

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. ■

The final result we will need is a very basic fact from numerical analysis controlling the error in a perturbed linear system:

Theorem 10 *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*

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

The proof of a result like this can be found in most textbooks on numerical analysis (although it is more common to use the *condition number* of A rather than its smallest singular value). Since we are more interested in the $\|\cdot\|_\infty$ measure of distance, we give the following simple corollary:

Corollary 11 *Let A be a nonsingular $k \times k$ matrix, b be a k-dimensional vector, and x the solution to $Ax = b$. Assume that $\|x\|_\infty \leq 1$. Suppose A' is a $k \times k$ matrix such that each entry of $A - A'$ is at most ϵ_{matrix} in magnitude, and assume that $\epsilon_{\text{matrix}} < \sigma_k(A)/2k$. Let b' be a k-dimensional vector satisfying $\|b - b'\|_\infty \leq \epsilon_{\text{rhs}}$. Let x' be the solution to $A'x' = b'$. Then we have*

$$\|x - x'\|_\infty \leq O(k) \frac{\epsilon_{\text{matrix}} + \epsilon_{\text{rhs}}}{\sigma_k(A)}.$$

C Proof of Theorem 4

We go through the algorithm step by step, as it appears in Appendix A. In Step 1 of WAM, we define constants $\epsilon_{\text{wts}} = \epsilon^3$, $\tau = \epsilon^2/n^2$, and $\epsilon_{\text{matrix}} = \tau^{k+1}$, which we use throughout the proof.

In Step 2 of WAM the algorithm will grid over estimates $\hat{\pi}^i$ that satisfy $|\hat{\pi}^i - \pi^i|$ for all i. In this case, any mixing component $\mathbf{X}^i$ whose mixing weight π^i is at least ϵ will not be eliminated. Since we need not be concerned with accuracy for the means of the other mixing components, we can ignore them and assume for the rest of the proof that $\pi^i \geq \epsilon$ for all i.

Now we come to the main work in the proof of correctness of Theorem 4: namely, showing that in Steps 3–7 of algorithm WAM, accurate estimates for the $\tilde{\mu}_j^i$'s are produced. Our goal for most of the remainder of the proof will be to show we obtain estimates $\hat{\tilde{\mu}}_j^i$ satisfying

$$|\hat{\tilde{\mu}}_j^i - \tilde{\mu}_j^i| \leq \tilde{\epsilon} := \epsilon^2$$

for all i.

To that end, let $r^* = r_\tau^*(M)$, the τ-essential rank of M. We will quickly dismiss the two easy cases, $r^* = 0$ and $r^* = k$; we then treat the general case $0 < r^* < k$.

$r^* = 0$ case. By definition, in this case $\sigma_1(M) \le \tau \le \tilde{\epsilon}$. Since $\sigma_1(M)$ is at least as large as the magnitude of M's largest entry we must therefore have $|\tilde{\mu}^i_j| \le \tilde{\epsilon}$ for all i, j. Now when WAM tries $r^* = 0$ in Step 4, tries the k standard basis vectors for $\hat{u}_1, \dots, \hat{u}_k$ in Step 5, and chooses all of these vectors for $\mathcal{J}$ in Step 6, it will set $\widehat{B} = 0$ in Step 7 and get $\hat{\tilde{\mu}}^i_j = 0$ for all i, j when it solves the linear system. But this is indeed within an additive $\tau \le \tilde{\epsilon}$ of the true values, as desired.

$r^* = k$ case. By definition, it's not hard to see that in this case we must have $\sigma_k(M) \ge \tau^k$. Now consider when WAM tries $r^* = k$ in Step 4. Step 5 becomes vacuous. By Corollary 5 there is some set of k columns $\mathcal{J} = J$ such that $\sigma_k(M_{\mathcal{J}}) \ge \sigma_k(M)/\sqrt{k(n-k)+1} \ge \tau^k/n$. In Step 6 WAM will try out this $\mathcal{J}$ and grid the associated entries to within $\pm\epsilon_{\text{matrix}}$. In Step 7 the algorithm will use only $\widehat{\text{corr}}$'s in forming $\widehat{B}$ and these will also be correct to within an additive $\pm\epsilon_{\text{matrix}}$. We can now use Corollary 11 — note that $\epsilon_{\text{matrix}} = \tau^{k+1} \le (\tau^k/n)/2k \le \sigma_k(M_{\mathcal{J}})/2k$, as necessary. This gives estimates in Step 7 satisfying

$$|\hat{\tilde{\mu}}^i_j - \tilde{\mu}^i_j| \le O(k)\frac{2\epsilon_{\text{matrix}}}{\tau^k/n} = O(kn\tau) \le \tilde{\epsilon},$$

as desired.

$0 < r^* < k$ case. In this case, by definition of the essential rank, we have

$$\tau\sigma_{r^*}(M) \ge \sigma_{r^*+1}(M) \ge \tau^k. \tag{3}$$

In Step 4 WAM will try out the correct value for r^* and in Step 5 WAM will grid over vectors $\hat{u}_{r^*+1}, \dots, \hat{u}_k$ that are within $\pm\epsilon_{\text{matrix}}$ in each coordinate of the actual last left singular vectors of M, $u_{r^*+1}, \dots, u_k$. Let M' denote the matrix M with these true singular vectors adjoined. By Proposition 9 we have

$$\sigma_k(M') \ge \min\{1, \sigma_{r^*}(M)\}. \tag{4}$$

From the crude upper bound $\sigma_{r^*}(M) \le \|M\|_F = \sqrt{\sum_{i,j}(\tilde{\mu}^i_j)^2} \le \sqrt{kn}$, we can restate (4) as simply $\sigma_k(M') \ge \sigma_{r^*}(M)/\sqrt{kn}$. Now applying Corollary 5 we conclude there is a subset $\mathcal{J}$ of M''s columns with $|\mathcal{J}| = k$ such that

$$\sigma_k(M'_{\mathcal{J}}) \ge \sigma_k(M')/\sqrt{k(n-k)+1} \ge \sigma_{r^*}(M)/kn. \tag{5}$$

In Step 6, WAM will try this set of columns $\mathcal{J} = J \cup J'$; it will also grid estimates for the entries in this column that are correct up to an additive $\pm\epsilon_{\text{matrix}}$. Note that WAM now has an $\widehat{M'_{\mathcal{J}}}$ that has all entries correct up to an additive $\pm\epsilon_{\text{matrix}}$. Now consider the matrix $\widehat{B}$ WAM forms in Step 7. For the columns corresponding to J the entries are given by $\widehat{\text{corr}}$'s, which are correct to within $\pm\epsilon_{\text{matrix}}$. For the columns corresponding to J' the entries are 0's; by the second part of Proposition 9 these are correct up to an additive $\sigma_{r^*+1}(M)$. We now use Corollary 5 to bound the error resulting from solving the system $M^\top_{\mathcal{J}}\widehat{M'}_{\mathcal{J}} = \widehat{B}$ in Step 7. To check that the necessary hypothesis is satisfied we combine (3) and (5):

$$\sigma_k(M'_{\mathcal{J}})/2k \ge \sigma_{r^*}(M)/2k^2n \ge \tau^{k-1}/2k^2n \ge \tau^{k+1} = \epsilon_{\text{matrix}}.$$

Now Corollary 11 tells us that the $\hat{\tilde{\mu}}^i_j$ produced satisfy

$$|\hat{\tilde{\mu}}^i_j - \tilde{\mu}^i_j| \le O(k)\frac{\epsilon_{\text{matrix}} + \max\{\epsilon_{\text{matrix}}, \sigma_{r^*+1}(M)\}}{\sigma_k(M'_{\mathcal{J}})} \le O(k^2n)\frac{\epsilon_{\text{matrix}} + \sigma_{r^*+1}(M)}{\sigma_{r^*}(M)},$$

where in the last step we used (5). But by (3) we have $\epsilon_{\text{matrix}}/\sigma_{r^*}(M) \leq \epsilon_{\text{matrix}}/\tau^{k-1} = \tau^2$ and also $\sigma_{r^*+1}(M)/\sigma_{r^*}(M) \leq \tau$. Thus we have

$$|\hat{\tilde{\mu}}^i_j - \tilde{\mu}^i_j| \leq O(k^2 n)\tau \leq \tilde{\epsilon},$$

as desired.

It remains to bound the error blowup in Step 8. By this point we have values for the π^i's that are accurate to within $\pm\epsilon_{\text{wts}}$, and further, all π^i's are at least ϵ. We also have values for all $\tilde{\mu}^i_j$'s that are accurate to within $\pm\tilde{\epsilon}$. Since the function $g(x,y) = y/\sqrt{x}$ satisfies

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

the Mean Value Theorem implies that in Step 8 our resulting estimates $\hat{\mu}^i_j$ are accurate to within additive error

$$\epsilon_{\text{wts}} \cdot 2\epsilon^{-3/2} + \tilde{\epsilon} \cdot \epsilon^{-1/2} \leq \epsilon,$$

as necessary.

This completes the proof of WAM's correctness. As for the running time, it is easy to see that the dominating factor comes from gridding over the entries of M_J and $u_{r^*+1}, \dots, u_k$. Since there are k^2 entries and we grid to granularity $\epsilon_{\text{matrix}} = \tau^{k+1} = \text{poly}(n/\epsilon)^k$, the overall running time is $\text{poly}(n/\epsilon)^{k^3}$; i.e., $\text{poly}(n/\epsilon)$ for constant k. ∎

D Proof of Theorem 6

For each $\ell = 1, \dots, b-1$, the algorithm runs WAM on the random variable $\mathbf{Z}^\ell$. In each such run, the "ϵ" parameter of WAM is set to $\epsilon' := \epsilon\sigma_b/(O(b)\cdot(b-1)^{b-1})$, where σ_b is a constant we define later, and the "δ" parameter is set to $\delta' := \delta/(b-1)$. From these runs we obtain $(b-1)$ lists $L_1, \dots, L_{b-1}$ of candidates $\langle\{\hat{\pi}^i\}, \{\hat{\mu}^i_{j,\ell}\}_{i,j}\rangle$, where $\hat{\mu}^i_{j,\ell}$ is an estimate of $\mu^i_{j,\ell} = \mathbf{E}[(X^i_j)^\ell]$. The algorithm then uses these $(b-1)$ lists to construct one larger list L of candidates $\langle\{\hat{\pi}^i\}, \{\mu^i_{j,\ell}\}_{i,j,\ell}\rangle$, where each candidate estimates the mixing weights and all $b-1$ moments. This is done by taking all possible combinations of one candidate from each of the $b-1$ lists $L_1, \dots, L_{b-1}$, and combining them as follows: take the mixing weights $\{\hat{\pi}^i\}$ from the candidate from list L_1, and for $\ell = 1, \dots, b-1$, take $\{\mu^i_{j,\ell}\}_{i,j}$ from the candidate from list L_ℓ. The list L will have size $|L| = \prod_{\ell=1}^{b-1}|L_\ell| = \text{poly}(n, 1/\epsilon)$.

Theorem 4 on the WAM algorithm guarantees that with probability at least $1-(b-1)\delta' = 1-\delta$, each list L_ℓ contains a candidate whose $\{\hat{\mu}^i_{j,\ell}\}$ are accurate estimates of the ℓth moments. When we choose the accurate candidate from each list, we will obtain an overall candidate in L that is accurate on *all* $b-1$ *moments*. Define $\epsilon'' := \epsilon'(b-1)^{b-1}/2 = \epsilon\sigma_b/O(b)$. Formally, the list L will contain a candidate $\langle\{\hat{\pi}^i\}, \{\hat{\mu}^i_{j,\ell}\}_{i,j,\ell}\rangle$ such that **(i)** $|\hat{\pi}^i - \pi^i| \leq \epsilon''$ for all $i \in [k]$; and **(ii)** $|\hat{\mu}^i_{j,\ell} - \mu^i_{j,\ell}| \leq \epsilon''$ for all i, j, ℓ such that $\pi^i \geq \epsilon''$. (The extra factor of $(b-1)^{b-1}/2$ comes from the need to scale the distributions for WAM so that the means fall into the range $[-1,1]$.)

To complete the proof of the theorem, we must show how the algorithm converts each candidate $\langle\{\hat{\pi}^i\}, \{\hat{\mu}^i_{j,\ell}\}\rangle$ in the list L into "parametric" form $\langle\{\hat{\pi}^i\}, \{\hat{p}^i_{j,\ell}\}\rangle$ so that the "good" candidate satisfying **(i)** and **(ii)** above does not incur much error. It is easy to see that for a given $i \in [k], j \in [n]$, we have $(\mu^i_{j,0}, \dots, \mu^i_{j,b-1}) = (p^i_{j,0}, \dots, p^i_{j,b-1})V$, where V is a $b \times b$ Vandermonde matrix (more precisely, $V_{\alpha,\beta} = (\alpha-1)^{\beta-1}$, with $V_{1,1} = 1$.) Following this characterization, the algorithm computes

$(\hat{p}^i_{j,0}, \ldots, \hat{p}^i_{j,b-1}) = (\hat{\mu}^i_{j,0}, \ldots, \hat{\mu}^i_{j,b-1})V^{-1}$ for each i, j to obtain parametric estimates $\{\hat{p}^i_{j,\ell}\}$ for the probabilities $\{p^i_{j,\ell}\}$.

Now applying Corollary 11, we have that for all i, j, ℓ, we have $|\hat{p}^i_{j,\ell} - p^i_{j,\ell}| \le \epsilon'' \cdot O(b)/\sigma_b = \epsilon$, where σ_b is set equal to $\sigma_b(V)$, the smallest singular value of V. (Since the Vandermonde matrix is nonsingular, even without specifying σ_b we have that it is a positive constant that depends only on b; it can be shown to be at least $b^{-\mathrm{poly}(b)}$) The running time is dominated by the time to take the cross-product of the lists. This concludes the proof of Theorem 6. We remark that the running time dependence on b is of the form $(n/\epsilon)^{\mathrm{poly}(b)}$; since a b in the exponent is inevitable in our cross-product approach, we have refrained from excessive optimization of the dependence on b (by doing things such as representing the alphabet by bth roots of unity rather than equally spaced reals, which would have given a better Vandermonde singular value bound).

E The road ahead

Since the binary domain $\{0,1\}^n$ corresponds to the $b = 2$ case of the general $\{0, \ldots, b-1\}^n$ domain, here we shall deal only with the latter.

Recall that $p^i_{j,\ell}$ is the probability that under the ith product distribution over $\{0, \ldots, b-1\}^n$ in the target mixture $\mathbf{Z}$, the jth coordinate takes value ℓ. From Theorem 6, we have a list L of M *candidates* $\langle\{\hat{\pi}^i\}, \{\hat{p}^i_{j,\ell}\}\rangle$ such that at least one candidate is *parametrically accurate* — i.e., satisfies the following:

1. $|\hat{\pi}^i - \pi^i| \le \epsilon$ for all $i = 1 \ldots k$; and
2. $|\hat{p}^i_{j,\ell} - p^i_{j,\ell}| \le \epsilon$ for all $i \in [k], j \in [n]$ and $\ell \in \{0, \ldots, b-1\}$ such that $\pi^i \ge \epsilon$.

In Section F, we show how to convert candidate into a true mixture of product distributions, in such a way that any parametrically accurate candidate becomes a mixture distribution with small KL divergence from the target distribution (see Theorem 12). Applying this conversion procedure to the list from Theorem 6, we get a list of M hypothesis mixture distributions such that at least one hypothesis in the list has small KL divergence from the target $\mathbf{Z}$ (see Theorem 16).

Then in Section G we show how a maximum-likelihood procedure can find a KL-accurate hypothesis (one with small KL divergence from $\mathbf{Z}$) from among a list of hypothesis, one of which is guaranteed to have good KL divergence (see Theorem 17).

In Section H we combine Theorem 17 with Theorem 16 to obtain Theorem 2.

F From candidates to hypothesis mixture distributions

The following theorem defines a process that converts a single candidate for the π^i's and $p^i_{j,\ell}$'s of $\mathbf{Z}$ to a true mixture of product distributions over $\{0, \ldots, b-1\}^n$ that has at least some minimum mass on every point in $\{0, \ldots, b-1\}^n$ (as we will see in Section G, this minimum mass condition is required by the maximum-likelihood procedure). More importantly, the theorem guarantees that if the candidate is parametrically accurate then the process outputs a mixture distribution with small KL divergence relative to $\mathbf{Z}$.

Theorem 12

1. *There is an efficient procedure $\mathcal{A}$ which takes values $\epsilon_{\mathrm{bprobs}}, \epsilon_{\mathrm{wts}} > 0$ and $\hat{\pi}^i$,$\hat{p}^i_{j,\ell}$ as inputs and outputs a mixture $\dot{\mathbf{Z}}$ of k product distributions over $\{0, \ldots, b-1\}^n$ with mixing weights $\dot{\pi}^i > 0$ and probabilities $\dot{p}^i_{j,\ell} > 0$ satisfying*

   (a) $\sum_{i=1}^k \dot{\pi}^i = 1$, *and for each $i \in [k]$ and $j \in [n]$,* $\sum_{\ell=0}^{b-1} p^i_{j,\ell} = 1$;

(b) $\dot{\mathbf{Z}}(x) \geq (\epsilon_{\text{bprobs}})^n$ for all $x \in \{0, \ldots, b-1\}^n$.

2. *Furthermore, suppose* $\mathbf{Z}$ *is a mixture of* k *product distributions on* $\{0, \ldots, b-1\}^n$ *with mixing weights* $\pi^1, \ldots, \pi^k$ *and probabilities* $p^i_{j,\ell}$, *and that the following are satisfied:*

   (a) *for* $i = 1 \ldots k$ *we have* $|\pi^i - \hat{\pi}^i| \leq \epsilon_{\text{wts}}$, *and*

   (b) *for all* i, j, ℓ *such that* $\pi^i \geq \epsilon_{\text{minwt}}$ *we have* $|p^i_{j,\ell} - \hat{p}^i_{j,\ell}| \leq \epsilon_{\text{bprobs}}$.

   Then for sufficiently small ϵ_{bprobs} *and* ϵ_{wts}, *the mixture* $\dot{\mathbf{Z}}$ *will satisfy*

$$\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq \eta(\epsilon_{\text{bprobs}}, \epsilon_{\text{wts}}, \epsilon_{\text{minwt}}), \tag{6}$$

   where

$$\eta(\epsilon_{\text{bprobs}}, \epsilon_{\text{wts}}, \epsilon_{\text{minwt}}) := n \cdot (12b^3 \epsilon_{\text{bprobs}}^{1/2}) + k\epsilon_{\text{minwt}} n \ln(b/\epsilon_{\text{bprobs}}) + \epsilon_{\text{wts}}^{1/3}.$$

We prove Theorem 12 in Section F.2 after setting up the required machinery in Section F.1.

F.1 Some tools. Here we give some propositions which will be used in the proof of Theorem 12.

The following simple proposition bounds the KL divergence between two product distributions in terms of the KL divergences between their coordinates.

Proposition 13 *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 $n = 2$:

$$\begin{aligned}
\mathrm{KL}(\mathbf{P}_1 \times \mathbf{P}_2||\mathbf{Q}_1 \times \mathbf{Q}_2) &= \sum_x \sum_y \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_1(x)\mathbf{P}_2(y)}{\mathbf{Q}_1(x)\mathbf{Q}_2(y)} \\
&= \sum_x \sum_y \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_1(x)}{\mathbf{Q}_1(x)} + \sum_x \sum_y \mathbf{P}_1(x)\mathbf{P}_2(y) \ln \frac{\mathbf{P}_2(y)}{\mathbf{Q}_2(y)} \\
&= \sum \mathbf{P}_2(y)\mathrm{KL}(\mathbf{P}_1||\mathbf{Q}_1) + \sum \mathbf{P}_1(x)\mathrm{KL}(\mathbf{P}_2||\mathbf{Q}_2) \\
&\leq \epsilon_1 + \epsilon_2.
\end{aligned}$$

The general case follows by induction. ∎

Very roughly speaking, the following proposition states that if $\mathbf{P}$ is a π-weighted mixture of distributions $\mathbf{P}^1, \ldots, \mathbf{P}^k$ and $\mathbf{Q}$ is a γ-weighted mixture of distributions $\mathbf{Q}^1, \ldots, \mathbf{Q}^k$, then if each $\mathbf{Q}^i$ is "close" to the corresponding $\mathbf{P}^i$ and the π-weighting is "close" to the γ-weighting, then $\mathbf{Q}$ is "close" to $\mathbf{P}$. To make this precise we need several technical conditions as stated in the proposition.

Proposition 14 *Let* $\pi^1, \ldots, \pi^k, \gamma^1, \ldots, \gamma^k \geq 0$ *be mixing weights satisfying* $\sum \pi^i = \sum \gamma^i = 1$. *Let* $\mathcal{I} = \{i : \pi^i \geq \epsilon_3\}$. *Let* $\mathbf{P}^1, \ldots, \mathbf{P}^k$ *and* $\mathbf{Q}^1, \ldots, \mathbf{Q}^k$ *be distributions. Suppose that*

1. $|\pi^i - \gamma^i| \leq \epsilon_1$ *for all* $i \in [k]$;

2. $\gamma^i \geq \epsilon_2$ *for all* $i \in [k]$;

3. $\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) \leq \epsilon_{\mathcal{I}}$ *for all* $i \in \mathcal{I}$;

4. $\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) \leq \epsilon_{\text{all}}$ *for all* $i \in [k]$.

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

$$\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \le \epsilon_{\mathcal{I}} + k\epsilon_3\epsilon_{\mathrm{all}} + k\epsilon_4 \ln\frac{\epsilon_4}{\epsilon_2} + \frac{\epsilon_1}{\epsilon_4 - \epsilon_1}.$$

Proof:

$$\begin{aligned}
\mathrm{KL}(\mathbf{P}||\mathbf{Q}) &= \sum\Big(\sum_i \pi^i\mathbf{P}^i\Big)\ln\frac{\sum_i \pi^i\mathbf{P}^i}{\sum_i \gamma^i\mathbf{Q}^i} \\
&\le \sum\sum_i \pi^i\mathbf{P}^i \ln\frac{\pi^i\mathbf{P}^i}{\gamma^i\mathbf{Q}^i} \qquad\qquad \text{(by the log-sum inequality [7])} \\
&= \sum_i \pi^i \sum\Big(\mathbf{P}^i\ln\frac{\mathbf{P}^i}{\mathbf{Q}^i} + \mathbf{P}^i\ln\frac{\pi^i}{\gamma^i}\Big) \\
&= \sum_i \pi^i\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) + \sum_i \pi^i\ln\frac{\pi^i}{\gamma^i} \\
&= \left(\sum_{i\in\mathcal{I}} \pi^i\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i)\right) + \left(\sum_{i\notin\mathcal{I}} \pi^i\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i)\right) + \sum_i \pi^i\ln\frac{\pi^i}{\gamma^i}. \qquad (7)
\end{aligned}$$

For the first term of (7), we have

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

For the second term of (7), we have

$$\sum_{i\notin\mathcal{I}} \pi^i\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i) \;\le\; k\epsilon_3 \cdot \max_{i\in[k]}\{\mathrm{KL}(\mathbf{P}^i||\mathbf{Q}^i)\} \le k\epsilon_3\epsilon_{\mathrm{all}}.$$

For the third term of (7), letting $\mathcal{I}' = \{i \in \mathcal{I} : \pi^i \ge \epsilon_4\}$, we have

$$\sum_i \pi^i\ln\frac{\pi^i}{\gamma^i} = \sum_{i\notin\mathcal{I}'} \pi^i\ln\frac{\pi^i}{\gamma^i} + \sum_{i\in\mathcal{I}'} \pi^i\ln\frac{\pi^i}{\gamma^i}. \qquad (8)$$

For the first sum in (8) we have

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

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

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

Hence for the second sum in (8), we have

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

Putting all the bounds together the proof is done. ■

Finally, we will also need the following elementary proposition:

Proposition 15 *Let* $\mathbf{P}$ *and* $\mathbf{Q}$ *denote distributions over* $\{0,\ldots,b-1\}$ *where* $\mathbf{P}$ *has probabilities* $p_0,\ldots,p_{b-1}$ *and* $\mathbf{Q}$ *has probabilities* $q_0,\ldots,q_{b-1}$*. Suppose that* $|p_\ell - q_\ell| < \xi \le \frac{1}{4}$ *for all* $\ell \in \{0,\ldots,b-1\}$*, and that also* $q_\ell \ge \tau$ *for all* $\ell \in \{0,\ldots,b-1\}$*, where* $\tau < \xi$*. Then* $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \le 2\xi^{1/2} + b\xi^{3/2}/\tau$*.*

Proof: Let $L_{small} = \{\ell \in \{0,\ldots,b-1\} : p_\ell \le \xi^{1/2}\}$ and $L_{big} = \{0,\ldots,b-1\} \setminus L_{small}$. We bound the contribution to $\mathrm{KL}(\mathbf{P}||\mathbf{Q})$ from L_{small} and L_{big} separately.

Now for the L_{small} case. For all ℓ, it is easy to see that $\ln \frac{p_\ell}{q_\ell} \le \ln \frac{\xi+\tau}{\tau} = \ln(1+\frac{\xi}{\tau}) \le \frac{\xi}{\tau}$. Thus each $\ell \in L_{small}$ contributes at most $p_\ell \ln \frac{p_\ell}{q_\ell} \le \frac{\xi^{3/2}}{\tau}$. Since $|L_{small}| \le b$ the total contribution to $\mathrm{KL}(\mathbf{P}||\mathbf{Q})$ from L_{small} is at most $b\frac{\xi^{3/2}}{\tau}$.

If $\ell \in L_{big}$, then we have

$$\frac{p_\ell}{q_\ell} \le \frac{p_\ell}{p_\ell - \xi} = 1 + \frac{\xi}{p_\ell - \xi} \le 1 + \frac{\xi}{\xi^{1/2} - \xi} \le 1 + 2\xi^{1/2}$$

where the last inequality holds since $\xi^{1/2} \le \xi^{1/2}/2$ (since $\xi \le \frac{1}{4}$). We thus have that the total contribution to $\mathrm{KL}(\mathbf{P}||\mathbf{Q})$ from $\ell \in L_{big}$ is at most $\ln(1+2\xi^{1/2}) \le 2\xi^{1/2}$. This proves the proposition. ■

F.2 Proof of Theorem 12. 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 probabilities $\dot{p}^i_{j,\ell}$. The procedure $\mathcal{A}$ is defined as follows:

1. For all $i = 1,\ldots,k$ let
$$\ddot{\pi}^i = \begin{cases} \hat{\pi}^i & \text{if } \hat{\pi}^i \ge \epsilon_{\text{wts}} \\ \epsilon_{\text{wts}} & \text{if } \hat{\pi}^i < \epsilon_{\text{wts}}. \end{cases}$$
Now let s be such that $s\sum_{i=1}^k \ddot{\pi}^i = 1$, and take $\dot{\pi}^i = s\ddot{\pi}^i$.

2. For all $i \in [k]$ and $j \in [n]$, let
$$\ddot{p}^i_{j,\ell} = \begin{cases} \hat{p}^i_{j,\ell} & \text{if } \hat{p}^i_{j,\ell} \ge \epsilon_{\text{bprobs}} \\ \epsilon_{\text{bprobs}} & \text{if } \hat{p}^i_{j,\ell} < \epsilon_{\text{bprobs}}. \end{cases}$$
Now let t be such that $t\sum_{\ell\in\{0,\ldots,b-1\}} \ddot{p}^i_{j,\ell} = 1$, and take $\dot{p}^i_{j,\ell} = t\ddot{p}^i_{j,\ell}$.

It is clear from construction that this yields $\dot{\pi}^i$, $\dot{p}^i_{j,\ell}$ that satisfy condition 1(a) of the theorem. It is also clear that for each $i \in [k]$ we have that the distribution $\dot{\mathbf{Z}}^i$ satisfies $\dot{\mathbf{Z}}^i(x) \ge \epsilon^n_{\text{bprobs}}$ for all $x \in \{0,\ldots,b-1\}^n$, and thus the mixture $\dot{\mathbf{Z}}$ must satisfy $\dot{\mathbf{Z}}(x) \ge \epsilon^n_{\text{bprobs}}$ for all x. This gives part 1(b) of the theorem.

We now turn to part 2, and henceforth assume that the conditions on $\pi^i, \hat{\pi}^i$, $p^i_{j,\ell}$, $\hat{p}^i_{j,\ell}$ from part 2 are indeed all satisfied. Roughly speaking, these conditions tell us that $\hat{\pi}^i$, $\hat{p}^i_{j,\ell}$ are "good" (in the sense that they are parametrically accurate); we will show that the resulting $\dot{\pi}^i$, $\dot{p}^i_{j,\ell}$ are "good" (in the sense of giving rise to a mixture $\dot{\mathbf{Z}}$ that satisfies (6)).

Our goal is to apply Proposition 14 with parameter settings

$$\epsilon_1 = 3k\epsilon_{\text{wts}};\ \epsilon_2 = \frac{\epsilon_{\text{wts}}}{2};\ \epsilon_3 = \epsilon_{\text{minwt}};\ \epsilon_4 = \epsilon^{1/2}_{\text{wts}};\ \epsilon_{\mathcal{I}} = 12nb^3\epsilon^{1/2}_{\text{bprobs}};\ \epsilon_{\text{all}} = n\ln(b/\epsilon_{\text{bprobs}}). \quad (9)$$

to bound $\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}})$. To satisfy the conditions of Proposition 14 we must (1) upper bound $|\pi^i - \dot{\pi}^i|$ for all i; (2) lower bound $\dot{\pi}^i$ for all i; (3) upper bound $\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i)$ for all i such that $\pi^i \ge \epsilon_{\text{minwt}}$; and (4) upper bound $\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i)$ for all $i \in [k]$. We now do this.

(1) Upper bounding $|\pi^i - \dot{\pi}^i|$. Fix any $i \in [k]$. If $\hat{\pi}^i \geq \epsilon_{\text{wts}}$ then we have $\ddot{\pi}^i = \hat{\pi}^i$ so $|\pi^i - \ddot{\pi}^i| \leq \epsilon_{\text{wts}}$. On the other hand, if $\hat{\pi}^i < \epsilon_{\text{wts}}$ then it must be the case that $\pi^i \leq 2\epsilon_{\text{wts}}$ so we again have $|\pi^i - \ddot{\pi}^i| \leq \epsilon_{\text{wts}}$. Since $\sum_{i=1}^k \pi^i = 1$ it follows that

$$\left| \sum_{i=1}^{k} \ddot{\pi}^i - 1 \right| \leq k\epsilon_{\text{wts}} \tag{10}$$

and thus

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

By definition of s this gives

$$s \in \left[\frac{1}{1 + k\epsilon_{\text{wts}}}, \frac{1}{1 - k\epsilon_{\text{wts}}} \right] \tag{11}$$

Multiplying inequality (10) by s, recalling that $s \sum_{i=1}^k \ddot{\pi}^i = 1$, and assuming $\epsilon_{\text{wts}} \leq 1/(2k)$, 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 $|\pi^i - \dot{\pi}^i| \leq 3k\epsilon_{\text{wts}}$.

(2) Lower bounding $\dot{\pi}^i$. To lower bound $\dot{\pi}^i$, we note that since $\ddot{\pi}^i \geq \epsilon_{\text{wts}}$ for all i, and assuming $\epsilon_{\text{wts}} \leq 1/k$, we have

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

where the first inequality follows from (11).

(3) Upper bounding $\mathrm{KL}(\mathbf{Z}^i || \dot{\mathbf{Z}}^i)$ for all i such that $\pi^i \geq \epsilon_{\text{minwt}}$. Fix an i such that $\pi^i \geq \epsilon_{\text{minwt}}$ and fix any $j \in [n]$. Let $\mathbf{P}$ denote the distribution over $\{0, \ldots, b-1\}$ with probabilities $p^i_{j,0}, \ldots, p^i_{j,b-1}$ and let $\mathbf{Q}$ denote the distribution over $\{0, \ldots, b-1\}$ with probabilities $\dot{p}^i_{j,0}, \ldots, \dot{p}^i_{j,b-1}$.

We first show that each $\dot{p}^i_{j,\ell}$ is close to $\hat{p}^i_{j,\ell}$ and thus also to $p^i_{j,\ell}$. This is done much as in (1) above. If $\hat{p}^i_{j,\ell} \geq \epsilon_{\text{bprobs}}$ then we have $\ddot{p}^i_{j,\ell} = \hat{p}^i_{j,\ell}$ so $|p^i_{j,\ell} - \ddot{p}^i_{j,\ell}| \leq \epsilon_{\text{bprobs}}$ (by condition 2(b) in the theorem statement). On the other hand, if $\hat{p}^i_{j,\ell} < \epsilon_{\text{bprobs}}$ then it must be the case that $p^i_{j,\ell} \leq 2\epsilon_{\text{bprobs}}$ so we again have $|p^i_{j,\ell} - \ddot{p}^i_{j,\ell}| \leq \epsilon_{\text{bprobs}}$. Since $\sum_{\ell=0}^{b-1} p^i_{j,\ell} = 1$ it follows that

$$\left| \sum_{\ell=0}^{b-1} \ddot{p}^i_{j,\ell} - 1 \right| \leq b\epsilon_{\text{bprobs}} \tag{12}$$

and thus

$$\sum_{\ell=0}^{b-1} \ddot{p}^i_{j,\ell} \in [1 - b\epsilon_{\text{bprobs}}, 1 + b\epsilon_{\text{bprobs}}].$$

By definition of t this gives

$$t \in \left[\frac{1}{1+b\epsilon_{\mathrm{bprobs}}}, \frac{1}{1-b\epsilon_{\mathrm{bprobs}}}\right] \tag{13}$$

Multiplying inequality (12) by t, recalling that $t\sum_{\ell=0}^{b-1}\ddot{p}^i_{j,\ell} = 1$, and assuming $\epsilon_{\mathrm{bprobs}} \leq 1/(2b)$, we obtain

$$|1-t| \leq tb\epsilon_{\mathrm{bprobs}} \leq \frac{b\epsilon_{\mathrm{bprobs}}}{1-b\epsilon_{\mathrm{bprobs}}} \leq 2b\epsilon_{\mathrm{bprobs}}.$$

Thus, we have

$$\begin{aligned}
|p^i_{j,\ell} - \dot{p}^i_{j,\ell}| &\leq |p^i_{j,\ell} - \ddot{p}^i_{j,\ell}| + |\ddot{p}^i_{j,\ell} - \dot{p}^i_{j,\ell}| \\
&\leq \epsilon_{\mathrm{bprobs}} + |\ddot{p}^i_{j,\ell} - \dot{p}^i_{j,\ell}| \\
&= \epsilon_{\mathrm{bprobs}} + |(1-t)\ddot{p}^i_{j,\ell}| \\
&\leq \epsilon_{\mathrm{bprobs}} + 2b\epsilon_{\mathrm{bprobs}}|\ddot{p}^i_{j,\ell}| \\
&\leq \epsilon_{\mathrm{bprobs}} + 2b\epsilon_{\mathrm{bprobs}};
\end{aligned}$$

certainly, this gives $|p^i_{j,\ell} - \dot{p}^i_{j,\ell}| \leq 3b\epsilon_{\mathrm{bprobs}}$.

Moreover, since $\ddot{p}^i_{j,\ell} \geq \epsilon_{\mathrm{bprobs}}$ for all ℓ and $\dot{p}^i_{j,\ell} = t\ddot{p}^i_{j,\ell}$ where $t > \frac{1}{2}$ (by (13) and $\epsilon_{\mathrm{bprobs}} \leq 1/b$), we also have $\dot{p}^i_{j,\ell} \geq \epsilon_{\mathrm{bprobs}}/2$. We may thus apply Proposition 15 to $\mathbf{P}$ and $\mathbf{Q}$ (taking $\tau = \epsilon_{\mathrm{bprobs}}/2$ and $\xi = 3b\epsilon_{\mathrm{bprobs}}$), and we obtain $\mathrm{KL}(\mathbf{P}||\mathbf{Q}) \leq 2(3b\epsilon_{\mathrm{bprobs}})^{1/2} + b(3b\epsilon_{\mathrm{bprobs}})^{3/2}/(\epsilon_{\mathrm{bprobs}}/2)$. Routine simplification gives that this is at most $12b^3\epsilon_{\mathrm{bprobs}}^{1/2}$. Each $\mathbf{Z}^i$ ($\dot{\mathbf{Z}}^i$ respectively) is the product of n such distributions $\mathbf{P}$ (distributions $\mathbf{Q}$ respectively) over $\{0,\dots,b-1\}$. Therefore, by Proposition 13, we have $\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i) \leq n\cdot(12b^3\epsilon_{\mathrm{bprobs}}^{1/2})$ for all i with $\pi^i \geq \epsilon_{\mathrm{minwt}}$.

(4) Upper bounding $\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i)$ **for all** $i \in [k]$. This is simple: fix any $i \in [k]$. Since we know that $\dot{\mathbf{Z}}^i(x) \geq \epsilon_{\mathrm{bprobs}}^n$ for all $x \in \{0,\dots,b-1\}^n$, we immediately have

$$\mathrm{KL}(\mathbf{Z}^i||\dot{\mathbf{Z}}^i) \leq -H(\mathbf{Z}^i) + \ln(1/(\epsilon_{\mathrm{bprobs}})^n) \leq n\ln(b/\epsilon_{\mathrm{bprobs}}),$$

where $H(\mathbf{X}) := \sum_x \mathbf{X}(x)\ln(1/\mathbf{X}(x))$ denotes the"entropy in nats" of the random variable $\mathbf{X}$.

We can now apply Proposition 14 with the parameter settings given by (9). Proposition 14 implies:

$$\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \quad\leq\quad n\cdot(12b^3\epsilon_{\mathrm{bprobs}}^{1/2}) + k\epsilon_{\mathrm{minwt}}n\ln(b/\epsilon_{\mathrm{bprobs}}) + \left[k\epsilon_{\mathrm{wts}}^{1/2}\ln\frac{\epsilon_{\mathrm{wts}}^{1/2}}{\epsilon_{\mathrm{wts}}/2} + \frac{3k\epsilon_{\mathrm{wts}}}{\epsilon_{\mathrm{wts}}^{1/2} - 3k\epsilon_{\mathrm{wts}}}\right].$$

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

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

and

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

(note that these inequalities only require that ϵ_{wts} is at most a sufficiently small constant depending only on k, roughly $1/k^6$).

Hence

$$\mathrm{KL}(\mathbf{Z}||\dot{\mathbf{Z}}) \leq n\cdot(12b^3\epsilon_{\mathrm{bprobs}}^{1/2}) + k\epsilon_{\mathrm{minwt}}n\ln(b/\epsilon_{\mathrm{bprobs}}) + \epsilon_{\mathrm{wts}}^{1/3}.$$

This concludes the proof of Theorem 12.

F.3 Some candidate distribution is good. Here we establish the following:

Theorem 16 *Let $b = O(1)$ and let $\mathbf{Z}$ be any unknown mixture of k product distributions over $\{0, \ldots, b-1\}^n$. There is a poly$(n/\epsilon) \cdot \log \frac{1}{\delta}$ time algorithm which, given samples from $\mathbf{Z}$, outputs a list of poly(n/ϵ) many mixtures of product distributions over $\{0, \ldots, b-1\}^n$ with the property that:*

- *every distribution $\mathbf{Z}'$ in the list satisfies $(\frac{\epsilon}{36nb^3})^{2n} \leq \mathbf{Z}'(x) \leq 1$ for all $x \in \{0, \ldots, b-1\}^n$; and*
- *with probability $1 - \delta$, some distribution $\mathbf{Z}^\star$ in the list satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$.*

Proof: We will use a specialization of Theorem 6 in which we have different parameters for the different roles that ϵ plays:

Theorem 6$'$: *Fix $k = O(1), b = O(1)$. Let $\mathbf{Z}$ be a mixture of k product distributions $\mathbf{X}^1, \ldots, \mathbf{X}^k$ over $\{0, \ldots, b-1\}^n$, so $\mathbf{Z}$ is described by mixing weights $\pi^1, \ldots, \pi^k$ and probabilities $\{p^i_{j,\ell}\}_{i \in [k], j \in [n], \ell \in \{0,\ldots,b-1\}}$.*

There is an algorithm with the following property: for any $\epsilon_{\mathrm{wts}}, \epsilon_{\mathrm{bprobs}}, \epsilon_{\mathrm{minwt}}, \delta > 0$, with probability $1-\delta$ the algorithm outputs a list of candidates $\langle \{\hat{\pi}^i\}, \{\hat{p}^i_{j,\ell}\}\rangle$ such that for at least one candidate in the list, the following holds:

1. *$|\hat{\pi}^i - \pi^i| \leq \epsilon_{\mathrm{wts}}$ for all $i \in [k]$; and*
2. *$|\hat{p}^i_{j,\ell} - p^i_{j,\ell}| \leq \epsilon_{\mathrm{bprobs}}$ for all i, j, ℓ such that $\pi^i \geq \epsilon_{\mathrm{minwt}}$.*

The algorithm runs in time poly$(n/\epsilon') \cdot \log(1/\delta)$, where $\epsilon' = \min\{\epsilon_{\mathrm{wts}}, \epsilon_{\mathrm{bprobs}}, \epsilon_{\mathrm{minwt}}\}$.

Let $\epsilon, \delta > 0$ be given. We run the algorithm of Theorem 6$'$ with parameters $\epsilon_{\mathrm{bprobs}} = (\frac{\epsilon}{36nb^3})^2$, $\epsilon_{\mathrm{minwt}} = \frac{\epsilon}{3kn \ln(1296 b^7 n^2/\epsilon^2)}$, and $\epsilon_{\mathrm{wts}} = \frac{\epsilon^3}{27}$. With these parameters the algorithm runs in time poly$(n/\epsilon) \cdot \log \frac{1}{\delta}$. By Theorem 6$'$, we get as output a list of poly(n/ϵ) many candidate parameter settings $\langle \{\hat{\pi}^i\}, \{\hat{\mu}^i_j\}\rangle$ with the guarantee that with probability $1 - \delta$ at least one of the settings satisfies

- $|\pi^i - \hat{\pi}^i| \leq \epsilon_{\mathrm{wts}}$ for all $i \in [k]$, and
- $|\hat{p}^i_{j,\ell} - p^i_{j,\ell}| \leq \epsilon_{\mathrm{bprobs}}$ for all i, j, ℓ such that $\pi^i \geq \epsilon_{\mathrm{minwt}}$.

We now pass each of these candidate parameter settings through Theorem 12. It follows that the resulting distributions each satisfy $\epsilon^n_{\mathrm{bprobs}} = (\frac{\epsilon}{36nb^3})^{2n} \leq \mathbf{Z}'(x) \leq 1$ for all $x \in \{0,1\}^n$. A routine verification shows that with our choice of $\epsilon_{\mathrm{bprobs}}$, $\epsilon_{\mathrm{minwt}}$ and ϵ_{wts} we have

$$n \cdot (12 b^3 \epsilon^{1/2}_{\mathrm{bprobs}}) \leq \frac{\epsilon}{3}, \quad k \epsilon_{\mathrm{minwt}} n \ln \frac{b}{\epsilon_{\mathrm{bprobs}}} \leq \frac{\epsilon}{3}, \quad \text{and} \quad \epsilon^{1/3}_{\mathrm{wts}} \leq \frac{\epsilon}{3}.$$

Thus $\eta(\epsilon_{\mathrm{bprobs}}, \epsilon_{\mathrm{wts}}, \epsilon_{\mathrm{minwt}}) \leq \epsilon$, and we have that at least one of the resulting distributions $\mathbf{Z}^\star$ satisfies $\mathrm{KL}(\mathbf{Z}||\mathbf{Z}^\star) \leq \epsilon$. ■

G Finding a good hypothesis using maximum likelihood

Theorem 16 gives us a list of distributions at least one of which is close to the target mixture distribution $\mathbf{Z}$ that 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 [14]) and we emphasize that the ideas behind Theorem 17 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

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

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 $\arg\max_{\mathbf{Q} \in \mathcal{Q}} \prod_{x \in S} \mathbf{Q}(x)$.

Theorem 17 *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 $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^{\mathrm{ML}}) \leq 4\epsilon$.

Before proving Theorem 17 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

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

where $H(\mathbf{P}) = -\sum_{x \in X} \mathbf{P}(x) \ln \mathbf{P}(x)$ is the "entropy in nats" of $\mathbf{P}$.

Consider the random variable $-\ln \mathbf{Q}(x)$, 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:

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

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

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

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 [16]:

Theorem 18 *(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 17.

Proof of Theorem 17: Call a distribution $\mathbf{Q} \in \mathcal{Q}$ *good* if $\mathrm{KL}(\mathbf{P}||\mathbf{Q}^{\mathrm{ML}}) \le 4\epsilon$, 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 $\Lambda(\mathbf{Q}) \le m(H(\mathbf{P})+3\epsilon)$ or the good distribution $\mathbf{Q}^*$ has $\Lambda(\mathbf{Q}^*) \ge m(H(\mathbf{P})+2\epsilon)$. Thus, by a union bound, we have

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

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

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

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

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

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

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

H Putting it all together

All the pieces are now in place for us to prove our main learning result, Theorem 2, for learning mixtures of product distributions over $\{0,\dots,b-1\}^n$.

Proof of Theorem 2: Run the algorithm described in Theorem 16. With probability $1-\delta$ this produces a list of $T=\mathrm{poly}(n/\epsilon)$ many hypothesis distributions, one of which has KL divergence at most ϵ from $\mathbf{Z}$ and each of which puts weight at least $(\frac{\epsilon}{36nb^3})^{2n}$ on every point in $\{0,\dots,b-1\}^n$. Now run the ML algorithm with $\alpha = (\frac{\epsilon}{36nb^3})^{2n}$, $\beta = 1$, and $m = \mathrm{poly}(n, 1/\epsilon)\ln(T/\delta)$. By Theorem 17, with probability at least $1-\delta$ the ML algorithm 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 $\mathrm{poly}(n/\epsilon)\cdot\log(1/\delta)$. Replacing ϵ by $\epsilon/4$ and δ by $\delta/2$ we are done. ■

Tracing through the proofs, it is easy to check that the running time dependence on k is $(n/\epsilon)^{O(k^3)} \cdot \log\frac{1}{\delta}$.

I Proof of Theorem 7

The following claim is used in the proof of Theorem 7:

Claim 19 *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 $i = 1, \dots, n$. It follows that the uniform distribution over S is a mixture of ℓ product distributions of this sort. ■

Theorem 7: *For any function $k(n)$, if there is a poly(n/ϵ) time algorithm which learns a mixture of $k(n)$ product distributions over $\{0,1\}^n$, then there is a poly(n/ϵ) 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 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 L_1-norm accuracy $1 - \frac{\epsilon}{2}$ and confidence $1 - \frac{\delta}{3}$. Recall from Section 1.1 that to learn to L_1-norm error ϵ it suffices to learn to KL-divergence ϵ^2, and thus we have that $M =$poly(n/ϵ) by our assumption on the running time of A.

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 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 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 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 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 19 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. [5] 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 [17] 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 7, the resulting algorithm for learning decision trees is easily seen to have an equivalent 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 $n^{\log k(n)}$.