On Learning Mixtures of Heavy-Tailed Distributions

Anirban Dasgupta John Hopcroft Jon Kleinberg Mark Sandler*

Abstract

We consider the problem of learning mixtures of arbitrary symmetric distributions. We formulate sufficient separation conditions and present a learning algorithm with provable guarantees for mixtures of distributions that satisfy these separation conditions. Our bounds are independent of the variances of the distributions; to the best of our knowledge, there were no previous algorithms known with provable learning guarantees for distributions having infinite variance and/or expectation.

For Gaussians and log-concave distributions, our results match the best known sufficient separation conditions [1, 15]. Our algorithm requires a sample of size $\tilde{O}(dk)$, where d is the number of dimensions and k is the number of distributions in the mixture. We also show that for isotropic power-laws, exponential, and Gaussian distributions, our separation condition is optimal up to a constant factor.

1. Introduction

Mixture models form one of the most fundamental classes of generative models for clustered data, and they are a basic topic of study in statistics and machine learning. The general problem of analyzing mixture models can be formulated as follows. There is a set of distributions $\boldsymbol{D}_1, \ldots \boldsymbol{D}_k$ in d dimensions, with relative *mixing weights* $w_1, \ldots w_k$. We do not see the distributions, but rather are given a sample $\mathcal{S}$ generated according to the following "mixture" process: to construct each individual sample point $\boldsymbol{s}$ the process randomly selects distribution $\boldsymbol{D}_i$ with probability w_i, and then it draws $\boldsymbol{s}$ from $\boldsymbol{D}_i$. The goal, given this sample, is to classify the points in the mixture, thereby approximately learning the underlying distributions. In this way, mixture models provide a very simple probabilistic framework for the problem of extracting clusters from data.

In the statistics literature, the canonical approach to analyzing mixture models is through a local search procedure known as *Expectation-Maximization (EM)*, which applies iterative improvement to arrive at estimates for the parameters of the distributions in the mixture [8, 12, 14]. This is an extremely general and flexible method, but it is known that the local optima found by the EM algorithm can be very far from the global optimum. In the theoretical computer science literature, on the other hand, there has been work on the learning of mixtures with provable guarantees [6, 1, 2, 15]. However, this line of work has focused on (and relied crucially on the properties of) distributions whose tails decay exponentially or faster, so that outliers are extremely rare.

The Present Work: Mixture Models with Arbitrary Distributions. Here we consider the question of whether provable guarantees can be obtained for algorithms that analyze mixture models with more general distributions, including those with heavy tails and with potentially infinite moments (including infinite variances or even infinite means). Such distributions arise naturally in a wide range of applications [13].

For a mixture of distributions that overlap very closely, it may be impossible to learn the individual distributions beyond a certain accuracy threshold. An important issue, therefore, is to understand the necessary *separation conditions* on the distributions, relating their overlap to the ability to distinguish between points from different distributions. Thus, there are three general open questions here.

- What separation is required to be able to correctly classify all but ε fraction of points with high probability?
- Can such learning be done with a sample of polynomial size?
- When can the learning of the mixture be carried out by a polynomial-time algorithm?

* Department of Computer Science, Cornell University, e-mail:{adg, hopcroft, kleinber, sandler}@cs.cornell.edu. The work of the third author was performed in part while on sabbatical leave at Carnegie Mellon University, supported by a David and Lucile Packard Foundation Fellowship and NSF grants CCF-0325453, IIS-0329064, and CCR-0122581.

For arbitrary distributions, very little is known for any of these questions.

We focus here on the first two questions, showing that for a broad class of distributions, we can learn mixtures *almost as well as* if we were given the precise density functions. Furthermore, the required sample complexity is almost linear in the dimension and the number of mixing components. For our most general results, we leave open the question of finding a polynomial-time algorithm.

Our main contribution is to present the first algorithm that provably learns arbitrary symmetric distributions with independent coordinates. The algorithm has the property that if the centers are sufficiently separated, then all but an ε fraction of points will be correctly classified with probability at least $(1-\delta)$. Second, we show that our separation is within a constant factor of that necessary for a broad class of distributions, including Gaussians, Laplacian and power-law. The required sample complexity is polynomial in d (the number of dimensions), k (the number of distributions), $1/w_{\min}$ (the smallest mixing weight), and $1/\varepsilon$. The running time depends on the separation and in the worst case is exponential in the number of dimensions.

Notation. We use bold symbols for vectors. For example $\mathbf{v}_i$ is the ith vector and v_i is the ith component of the vector $\mathbf{v}$. This rule applies to distributions as well, where bold $\boldsymbol{D}$ or $\boldsymbol{D}_i$ denotes a distribution in $\Re^d$, while $\mathcal{D}$ denotes one dimensional distribution. To denote a sample point s drawn from distribution $\mathcal{D}$, we write $s \in \mathcal{D}$.

Throughout the paper we consider two types of partitions. Specifically, we will consider a partition $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ of the coordinate set, which will be used for validation. Secondly, we partition the sample (or a subset) $\mathcal{S}$ into k groups, $\mathcal{C}_1, \ldots \mathcal{C}_k$ to represent classification results. To avoid confusion, we will consistently refer to the former as a partitioning and to the latter as a clustering. For a set of samples $\mathcal{C}$, we will write $\operatorname{med}\mathcal{C}$, to denote a point $\boldsymbol{\mu}$, such that μ_i is a median point of the ith coordinates of all samples in $\mathcal{C}$.

Given $\boldsymbol{s} \in \Re^d$, and a subset of its coordinates $X = \{i_1 \leq i_2 \leq \cdots \leq i_r\}$, we use $\boldsymbol{s}_X$ to denote the r-dimensional vector $(s_{i_1}, \ldots s_{i_r})$. Also, for an arbitrary partition $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ of the coordinate set, when it is clear from context, we will use $\boldsymbol{s}'$ and $\boldsymbol{s}''$ to denote $\boldsymbol{s}_{\mathcal{P}_1}$ and $\boldsymbol{s}_{\mathcal{P}_2}$ respectively.

Separation Conditions. In formulating these results, an important issue is the definition of the separation condition on the distributions. The recent work on mixtures of Gaussians has parameterized separation in terms of $\sigma_{\max}$, the maximum variance in any coordinate, and d, the number of dimensions. The initial work of Dasgupta used a random projection to learn distributions of Gaussians whose centers were at least $\Omega(\sigma_{\max}\sqrt{d})$ apart [6]. Soon thereafter, Dasgupta and Schulman [7] and Arora and Kannan[2] improved the required separation to $\Omega(d^{1/4}\sigma_{\max})$. The latter work also included an additional nonpositive term that allowed even concentric distributions, provided they have different variances. (This property did not carry through any of the subsequent results, including ours.) The separation condition was further improved by Vempala and Wang [15], who use spectral techniques [3] to learn mixtures of isotropic distributions. Their algorithm allowed a separation of $\tilde{\Omega}(\sigma_{\max})$,[1] and they noted that the logarithmic gap could be removed at the expense of a larger running time. The result of [15] was generalized to log-concave and non-isotropic distributions by Kannan et al [11] and Achlioptas and McSherry [1]. In the latter work, the class of distributions was further generalized to *g-concentrated and f-converged* and allowed limited dependence between coordinates, although these too have rapidly decaying tails.

When dealing with heavy-tailed distributions, the higher moments are less useful in defining separation conditions; indeed, they can be infinite. A useful principle in such cases is that medians can be more robust than means and moments. Motivated by this, we define the *median radius* of a one-dimensional distribution as follows, and subsequently parameterize the necessary separation conditions in these terms.

Definition 1 *Let X be a real random variable with cumulative density function $F(x)$. The* center *of X is the minimum c such that $F(c) = 1/2$. The* $\frac{1}{2}$-radius*, or* radius*, of X is the minimum value R such that half the probability mass lies in the interval $[c - R, c + R]$. A vector random variable $\mathbf{X}$, is said to have center at $\mathbf{c}$ and radius at most R, if each X_i has center at c_i and its median radius at most R.* [2].

We note here that basic tail inequalities imply the median radius is always at most $\sqrt{2}$ times the maximal variance. On the other hand, the variance might be much larger than the median radius; moreover, the median radius is defined for any distribution, even those with infinite variance.

To simplify the exposition of our results, we will only consider distributions that are symmetric around their centers, and with densities that monotonically decrease away from the centers. However, our results are easily

1 The $\tilde{\Omega}(.)$ notation is used to hide polylogarithmic factors.

2 Note that this median radius is computed for a single coordinate, as opposed to the d-dimensional median radius used in [2], which would be $O(\sqrt{d})$ times larger

generalized to distributions that have these properties only approximately.

There is a final notion, that we will use: this is the performance of the Bayes-optimal algorithm that knows all the parameters of the mixture model. Essentially, the performance of the Bayes-optimal algorithm is the best one could possibly achieve, and so it represents a useful baseline for comparison. For a broad class of distributions, we show that the separation conditions with which we can achieve strong learning results are necessary even for the Bayes-optimal algorithm to achieve good bounds. We note that approximation bounds with respect to Bayes-optimal date back to the seminal work of Cover and Hart [5], who showed that the nearest-neighbor algorithm is within a factor of two of the error rate of Bayes-optimal. However, their result required an exponential amount of *labeled data*, whereas our approach — based on a more complicated algorithm than the nearest-neighbor rule – does not use labeled data at all.

Overview of Results. Our results are concerned with two classes of high-dimensional distributions. The first class is $\mathcal{F}_0(R)$ consisting of all probability distributions in $\Re^d$ with independent coordinates, each with 1/2-radius at most R, and symmetric and monotonically decreasing tails. The second class is a subset of $\mathcal{F}_0$, denoted $\mathcal{F}_1(R)$. Any $\boldsymbol{D} \in \mathcal{F}_1(R)$, centered at $\boldsymbol{\mu}$, satisfies the additional condition that for any $x \in \mathcal{D}_i$, we have

$$\forall \alpha \geq 1, \Pr\left[|x - \mu_i| \geq \alpha R\right] \leq \frac{1}{2\alpha R}$$

We emphasize that this property is very weak. For example, all distributions with finite variance, as well as Zipf distributions with power coefficient at least one, satisfy it.

Recall that we assume a number of sample points that is polynomial in d, k, $1/w_{\min}$, and $1/\varepsilon$. We show that for a mixture of distributions $\boldsymbol{D}_1, \ldots \boldsymbol{D}_k$ from $\mathcal{F}_1$, with centers at $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_k$ satisfying the pairwise separation condition

$$||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega\left(\frac{Rk^{5/2}}{\varepsilon^2}\right)$$

there is an algorithm that correctly classifies all but an ε fraction of points with high probability.

For the more general case of distributions from $\mathcal{F}_0$, we need to impose a second type of condition as well, motivated by the following considerations. For any fixed separation it is possible to design two symmetric distributions in one dimension, with median radii equal to 1, such that any algorithm will misclassify points with probability at least 1/4. Suppose we now construct a d-dimensional distribution by using these one-dimensional distributions in each coordinate, and choosing centers that only differ in one coordinate. Then $n-1$ coordinates are providing no information, and in the remaining one coordinate we have a 1/4 probability of misclassification. Thus, to handle arbitrary distributions in $\mathcal{F}_0$, we need a *slope condition* that says, essentially, the centers are not aligned along one axis (or a small number of axes).

Specifically, we show that for a mixture of distributions $\boldsymbol{D}_1, \ldots \boldsymbol{D}_k$ from $\mathcal{F}_0$, with centers at $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_k$ satisfying $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega\left(R\sqrt{\frac{k}{\varepsilon}}\right)$ and $\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty} \geq \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ there is an algorithm that will correctly classify all but an ε fraction of points with high probability. The second requirement here is the specific form of the slope condition that we require.

The basic idea behind our approach is as follows. Suppose we knew the centers of each distribution; how would we classify them? An obvious answer would be to assign each point to the closest center. But which distance function should we use for defining "closeness"? We use the L_1 norm, and show a sense in which it is better for this purpose than L_2 norm. Again, this can be viewed as an application of a general robustness principle from statistics, that L_1 can be more robust than L_2 [13, 4, 10, 9]. However, we are not aware of previous applications of this principle to provide provable guarantees for mixture models of the type we obtain here.

If we are not given the centers, then the next idea is to find a reasonable estimate for them. We do this by exhaustive clustering of a subset $\mathcal{S}_0$ of size $\tilde{\Omega}(dk)$. To discriminate between correct and incorrect assignments, we develop a validation test that fails with a probability that is much smaller than $k^{-|\mathcal{S}_0|}$, and thus one could apply union bound. The idea behind the validation test is simply to partition the set of coordinates into two parts, and cluster on each subset independently. The validation fails if the two clusterings differ significantly. Finally we show that at the moment we consider the correct clustering, our estimate of the centers will be good enough that assigning points to the closest center works well.

2. Classification with known centers

Here we show that the algorithm that assigns each point to the closest center (with respect to L_1) works for arbitrary distributions. This results follows from the lemma below, which states that a sample drawn from symmetric distribution is much more likely to be closer to the center of the distribution, than to a fixed point

$\boldsymbol{\mu}$, given that $\boldsymbol{\mu}$ is sufficiently separated from center. Where separation includes both distance and slope conditions. We then eliminate the slope condition at the expense of narrowing the class of allowed distributions to $\mathcal{F}_1$.[3]

Lemma 2.1 *Let ε and C be constants and let $\boldsymbol{D}$ be a distribution centered at the origin with radius R. Let $\boldsymbol{\mu}$ be a point such that $||\boldsymbol{\mu}||_2 \geq 4R(C+\frac{1}{\sqrt{\varepsilon}})$, and having a slope ratio $\frac{||\boldsymbol{\mu}||_2}{||\boldsymbol{\mu}||_\infty}$ at least $4(C+\frac{1}{\sqrt{\varepsilon}})$. A point $\mathbf{x}$ sampled from $\boldsymbol{D}$ will satisfy*

$$||\mathbf{x}-\boldsymbol{\mu}||_1 - ||\mathbf{x}||_1 \geq C||\boldsymbol{\mu}||_2 \geq C^2 R \tag{1}$$

with probability at least $1-\varepsilon$.

Proof. We need to show that $Q = ||\mathbf{x}-\boldsymbol{\mu}||_1 - ||\mathbf{x}||_1 > C||\boldsymbol{\mu}||_2$ with probability at least $(1-\varepsilon)$. We write Q in the coordinate form

$$Q = \sum_{i=1}^{d} |x_i - \mu_i| - |x_i| = \sum_{i=1}^{d} q_i \tag{2}$$

where $q_i = |x_i - \mu_i| - |x_i|$. Recall that by Chebyshev's inequality, $\Pr[Q \leq \mathbf{E}[Q] - t\sigma(Q)] < \frac{1}{t^2}$. Since the absolute value of q_i is at most μ_i and since all q_i are independent, $\sigma(Q) \leq ||\boldsymbol{\mu}||_2$. It is sufficient then to properly estimate $\mathbf{E}[Q]$. Without loss of generality, assume that $\mu_i \geq 0$ for all i. Then

$$q_i = \begin{cases} \mu_i & x_i < 0 \\ \mu_i - 2x_i & 0 \leq x \leq \mu_i \\ -\mu_i & x_i \geq \mu_i \end{cases}. \tag{3}$$

Thus

$$\mathbf{E}[q_i] = \mu_i(\Pr[x_i \leq 0] - \Pr[x_i \geq \mu_i]) + \int_0^{\mu_i} (\mu_i - 2x)\mathcal{D}_i(x)dx$$

where $\mathcal{D}_i(x)$ is the density function of x_i. Using the fact that $\mathcal{D}_i$ is symmetric around 0 and non-increasing on $x \geq 0$ we obtain:

$$\mathbf{E}[q_i] \geq \mu_i \Pr[0 \leq x_i \leq \mu_i].$$

To estimate $\Pr[0 \leq x_i \leq \mu_i]$ we recall that $R \leq ||\boldsymbol{\mu}||_2/(4(C+\frac{1}{\sqrt{\varepsilon}}))$, and so at least half of the weight is concentrated in $[-\frac{||\boldsymbol{\mu}||_2}{4(C+\frac{1}{\sqrt{\varepsilon}})}, \frac{||\boldsymbol{\mu}||_2}{4(C+\frac{1}{\sqrt{\varepsilon}})}]$. Taking into account that $\mu_i \leq ||\boldsymbol{\mu}||_\infty \leq \frac{||\boldsymbol{\mu}||_2}{4(C+\frac{1}{\sqrt{\varepsilon}})}$ and that x_i is symmetric with decreasing density, we immediately have:

$$\Pr[0 \leq x_i \leq \mu_i] \geq \frac{\mu_i}{4} \frac{4(C+\frac{1}{\sqrt{\varepsilon}})}{||\boldsymbol{\mu}||_2} = \frac{\mu_i}{||\boldsymbol{\mu}||_2}(C+\frac{1}{\sqrt{\varepsilon}})$$

and so

$$\mathbf{E}\left[\sum_i q_i\right] \geq \sum_i \frac{\mu_i^2}{||\boldsymbol{\mu}||_2}(C+\frac{1}{\sqrt{\varepsilon}}) \geq ||\boldsymbol{\mu}||_2(C+\frac{1}{\sqrt{\varepsilon}}),$$

Thus, from Chebyshev's inequality we get

$$\Pr\left[Q \leq ||\boldsymbol{\mu}||_2((C+\frac{1}{\sqrt{\varepsilon}})) - t||\boldsymbol{\mu}||_2\right] \leq 1/t^2.$$

Substituting $t = \frac{1}{\sqrt{\varepsilon}}$, we get $\Pr[Q < C||\boldsymbol{\mu}||_2] \leq \varepsilon$. The final step that $C||\boldsymbol{\mu}||_2 \geq C^2 R$ follows from the condition $||\boldsymbol{\mu}||_2 \geq 4R(C+\frac{1}{\sqrt{\varepsilon}})$. ■

The slope condition in this lemma is a property of the L_1 norm, rather than the analysis. Consider the following example with two distributions: one is centered at the origin and the other is centered at $\boldsymbol{\mu} = (1000, 1, \ldots, 1)$. The coordinate density function of the first distribution has half of its mass uniformly distributed in the interval $[-0.01, 0.01]$ and the other half uniformly distributed over the remainder of the interval $[-10^{100}, 10^{100}]$. The second distribution is obtained from the first by translation by vector $\boldsymbol{\mu}$. Thus for any point, roughly half the coordinates will be very close to the center, but the other half will be almost uniformly distributed over a interval of size 2×10^{100}. Given this fact, the distributions are easy to distinguish. But our L_1 algorithm will fail to distinguish between the distributions with probability at least 1/5. This is because, in the computation of L_1 distance, the contribution of each coordinate is proportional to the distance between the centers on that coordinate, in this case to μ_i. Since the first coordinate is 1000 times any other coordinate, if the dimension $d < 1000$, then the correctness of the algorithm depends entirely on the contribution of the first coordinate. Since the first coordinate will wrong about one quarter of the time, the L_1 algorithm will be wrong with probability at least 1/5. This implies that our algorithm has a seemingly counter-intuitive property: For some instances of the problem, it is possible to move centers so that each coordinate difference increases, yet, as the slope ratio decreases, the probability of a wrong assignment increases. Observe that if $\boldsymbol{\mu} = (1, 1, \ldots 1)$ the L_1 algorithm would have worked correctly.

However, if we limit ourselves to the class $\mathcal{F}_1$ of distributions, then a slightly increased separation condition precludes this problem. Recall that $\boldsymbol{D}$ belongs to $\mathcal{F}_1(\boldsymbol{\mu}, R)$, if each $\mathcal{D}_i$ is symmetric around μ_i and for $x_i \in \mathcal{D}_i$ we have

$$\Pr[|x_i - \mu_i| \geq \alpha R] \leq \frac{1}{2\alpha}, \forall \alpha \geq 1. \tag{4}$$

The following lemma analyzes the classification rule for this class of distributions.

3 We remind the reader that $\mathcal{F}_1$ still includes most if not all of the standard distributions used for data analysis.

Lemma 2.2 *Fix $\varepsilon \leq \frac{1}{10}$. Suppose $\boldsymbol{D}_1 \in \mathcal{F}_1(R)$ and $\boldsymbol{\mu} \in \Re^d$ satisfies $||\boldsymbol{\mu}||_2 \geq \frac{6000R}{\varepsilon^2}$. Then $\mathbf{x}$ sampled from $\boldsymbol{D}_1$ will satisfy*

$$\Delta = ||\mathbf{x} - \boldsymbol{\mu}||_1 - ||\mathbf{x}||_1 \geq \frac{||\boldsymbol{\mu}||_2}{15}$$

with probability at least $1 - \varepsilon$.

Proof Sketch. To prove this lemma we split the set of coordinates into 'large' and 'small' coordinates according to the absolute values of μ_i, with threshold at $O(\frac{R}{\varepsilon})$. For the former set we use our tail condition (4) and for the latter set we use the technique from lemma 2.1. Further details are provided in the full version of the paper. ■

Lemmas 2.1 and 2.2, together with the union bound, make the following theorem immediate:

Theorem 2.3 *Consider a mixture of k distributions $\boldsymbol{D}_1, \ldots \boldsymbol{D}_k$, with known centers $\boldsymbol{\mu}_1, \ldots \boldsymbol{\mu}_k$. If either of the following conditions is satisfied, then classification according to the nearest center in the L_1 norm fails with probability at most ε.*

- *For every i and j, $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega(R\sqrt{\frac{k}{\varepsilon}})$ and $\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty} \geq \Omega(\sqrt{\frac{k}{\varepsilon}})$ or*
- *Each distribution belongs to class $\mathcal{F}_1$ and for every i and j, we have $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega(R\frac{k^2}{\varepsilon^2})$*

L_2 vs. L_1: an example. One might wonder why not to use L_2 norm to define closeness. Here we construct an example of a mixture of distributions which is separable by our L_1 algorithm but not using the L_2 norm. The data consists of two mixtures of Cauchy distributions: $\boldsymbol{D}_0$, centered at 0, and $\boldsymbol{D}_1$, centered at $\boldsymbol{\mu}$. The coordinate density functions of the one-dimensional Cauchy distributions has the form $\mathcal{D}_{0i}(x) = \frac{2/\pi}{x^2+1}$ and $\mathcal{D}_{1i}(x) = \frac{2/\pi}{(\mu_i - x)^2+1}$ We will show that with R/δ separation between the centers, any similar algorithm based on L_2 norm must make an error on a $Theta(1)$ fraction of the points, where the constant is independent of δ.

Note that the $1/2$ radius of the coordinate distributions is 1, thus L_1 by the theorem 2.3 L_1 norm classification works. For L_2 norm we prove the following:

Lemma 2.4 *Suppose $\boldsymbol{\mu} = \left(\frac{1}{\sqrt{\delta d}} \ldots, \frac{1}{\sqrt{\delta d}}\right)$.. For any constant δ, and for d sufficiently large, the algorithm which assigns sample $\mathbf{x}$ to $\tilde{\mathcal{C}}_0$ if $||\mathbf{x}||_2 \leq ||\mathbf{x} - \boldsymbol{\mu}||_2$, and to $\tilde{\mathcal{C}}_1$ otherwise misclassifies at least $1/4$ fraction of the points with high probability.*

The details of the proof are in the full version. The basic intuition can be gotten by looking at the random variable $Z = \|\mathbf{x} - \boldsymbol{\mu}\|_2^2 - \|\mathbf{x}\|_2^2 = \boldsymbol{\mu}^t\boldsymbol{\mu} - 2\mathbf{x}^t\boldsymbol{\mu}$. Note that Z follows a Cauchy distribution with scale parameter $2||\boldsymbol{\mu}||_1$, and center either at $||\boldsymbol{\mu}||_2^2$ or $-||\boldsymbol{\mu}||_2^2$. It easily follows that the probability of misclassification, which is the probability that Z is greater than (or less than) 0, is very close to $1/2$.

3. Algorithm for learning mixtures

In this section we present our main algorithm. In order to build some intuition, suppose again that we knew the centers. Theorem 2.3 from the previous section then tells us that assigning each point to the nearest center in the L_1 sense produces the correct clustering. Now, unless centers are aligned along one (or just a few) coordinate axes, if we partition the set of coordinates in half and cluster the points independently using each half of the coordinates, we should get approximately the same clusterings. This suggests selecting a sample $\mathcal{S}_0$ of the points, and keeping a test set $\mathcal{S}_1$ for cross-validation. We then exhaustively test all possible clusterings of $\mathcal{S}_0$. Let $\tilde{\mathcal{C}}_1, \ldots, \tilde{\mathcal{C}}_k$ be one such possible clustering of $\mathcal{S}_0$. For each of the clusters $\tilde{\mathcal{C}}_i$, the center $\tilde{\boldsymbol{\mu}}_i = \text{med}(\tilde{\mathcal{C}}_i)$ is computed. In order to do cross-validation, we first do a random partitioning of the coordinates into two sets $(\mathcal{P}_1, \mathcal{P}_2)$. The projection of the computed centers $\tilde{\boldsymbol{\mu}}_i$ onto $\mathcal{P}_1$ and $\mathcal{P}_2$ induce two clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ of the test set $\mathcal{S}_1$. These two clusterings can then be tested against one another to see if they match. For any sample $\mathbf{x}$, the probability of assigning $\mathbf{x} \in \boldsymbol{D}_i$ to a cluster in $\tilde{\mathcal{C}}'_j$ or $\tilde{\mathcal{C}}''_j$ depends only on the distribution that it has come from, and the two decisions are independent. Thus two clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ will be close to each other if and only if these probabilities are close to each other *and* all the probabilities are close to either 0 or 1. But then both $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are close to the true clustering and thus the loop will only be broken when both clusterings are correct.

In order for the cross-validation phase to work, we need that the centers are not aligned along only a few axes. To simplify our presentation, we start with an algorithm that assumes this condition holds. After that, we show that if all mixture components belong to $\mathcal{F}_1$, then the data can be split into "superclusters" each containing one or more distributions, so that within each "supercluster" this assumption is satisfied.

The algorithm without preprocessing is robust in the following sense: independently of the center location, if a certain clustering is accepted, then it is very close to the original one with high probability. If the centers are aligned, however, the algorithm might not find any clustering to be acceptable.

One further construction needs comment here. Instead of dividing $\mathcal{S}_0$ into k clusters, the algorithm ac-

tually divides it into $k+1$ clusters, making sure that the $k+1^{st}$ cluster $\tilde{\mathcal{C}}_{k+1}$ is *small*, and is then removed from the data set. We will see that this is done to handle errors introduced by the preprocessing phase. The intuition is that $\tilde{\mathcal{C}}_{k+1}$ will capture samples that are introduced in error by the preprocessing phase.

Following the algorithm, the analysis consists of three parts. First we show that partitioning the coordinates will change distances by at most a factor of 2 with probability at least $1-\frac{1}{k^2}$. Second, we show that if partitioning doesn't change the distances much, then when the algorithm considers the clustering $\tilde{\mathcal{C}}_i = \mathcal{S}_0 \cap \mathcal{C}_i$, with high probability it will terminate with an approximately correct classification. Finally we show that unless clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are approximately correct the algorithm will never declare success for any partition of the coordinates.

Algorithm 1

Input: Sample $\mathcal{S}$

Output: Clustering $(\tilde{\mathcal{C}}_1, \ldots, \tilde{\mathcal{C}}_k)$.

Description:

1. *Pick a random partition* $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2\}$ *of the coordinate set. For any vector* $\mathbf{v}$*, we will use* v' *and* v'' *to denote the projection to* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *respectively.*
2. *Pick a random subset* $\mathcal{S}_0 \subset \mathcal{S}$ *of size* $\frac{96dk\log\frac{dk}{8\delta}}{w_{\min}}$*, and let* $\mathcal{S}_1 = \mathcal{S} - \mathcal{S}_0$
3. *For all possible clusterings of* S_0 *into* $k+1$ *groups* $\{\tilde{\mathcal{C}}_1, \ldots \tilde{\mathcal{C}}_k, \tilde{\mathcal{C}}_{k+1}\}$*, do the following:*
   - 3a. *Check that group* $\tilde{\mathcal{C}}_{k+1}$ *contains less than* $\frac{\varepsilon|\mathcal{S}_0|}{2}$ *points, and ignore* $\tilde{\mathcal{C}}_{k+1}$ *in steps* **3b - 3e**
   - 3b. *For each group* $\tilde{\mathcal{C}}_i$*, compute* $\tilde{\boldsymbol{\mu}}_i = \operatorname{med}\tilde{\mathcal{C}}_i$*. Let* $\tilde{\boldsymbol{\mu}}_i'$ *and* $\tilde{\boldsymbol{\mu}}_i''$ *denote the projection of* $\tilde{\boldsymbol{\mu}}_i$ *into* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *respectively.*
   - 3c. *Cluster points from* $\mathcal{S}_1$ *with respect to* $\tilde{\boldsymbol{\mu}}'$ *and* $\tilde{\boldsymbol{\mu}}''$*. E.g:* $\tilde{\mathcal{C}}_i' = \{\boldsymbol{s} \in \mathcal{S}_1 \big| \forall j \in [1\ldots k],\ ||\boldsymbol{s}' - \tilde{\boldsymbol{\mu}}_i'||_1 \leq ||\boldsymbol{s}' - \tilde{\boldsymbol{\mu}}_j'||_1\}$ *and* $\tilde{\mathcal{C}}_i'' = \{\boldsymbol{s} \in \mathcal{S}_1 \big| \forall j \in [1\ldots k],\ ||\boldsymbol{s}'' - \tilde{\boldsymbol{\mu}}_i''||_1 \leq ||\boldsymbol{s}'' - \tilde{\boldsymbol{\mu}}_j''||_1\}$*.*
   - 3d. *If* $\sum_i \tilde{\mathcal{C}}_i' \triangle \tilde{\mathcal{C}}_i'' > 10\varepsilon m$ *or any of the clusters* $\tilde{\mathcal{C}}_i'$ *has size less than* $\frac{\varepsilon|S_1|}{2}$*, go to the next iteration.*
   - 3e. *Else, the* $\tilde{\mathcal{C}}_i'$ *that has been found corresponds to an approximately correct clustering. Set* $\tilde{\boldsymbol{\mu}}_i = \operatorname{med}\tilde{\mathcal{C}}_i'$*, and reassign all points from* $\mathcal{S}$ *to the closest* $\tilde{\boldsymbol{\mu}}_i$*. Stop the algorithm.*
4. *If this point is reached, repeat steps* $\mathbf{1}-\mathbf{3}$ *(up to a maximum of* $\log\frac{1}{\delta}$ *repetitions).*

3.1. Algorithm analysis

In all following results, we will be implicitly assuming that our centers satisfy one of the two following separation conditions:

- For every i and j, $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq 10R\sqrt{\frac{k}{\varepsilon}}$ *and* $\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty} \geq 10\sqrt{\frac{k}{\varepsilon}}$ *or*
- Each distribution belongs to class $\mathcal{F}_1$ *and* for every i and j, $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq 15000R\frac{k^2}{\varepsilon^2}$

Coordinate partitioning. We start with the definition of a "good partition."

Definition 2 *A partition* $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ *is called a* good partition *if, for all* i *and* j*, projection onto* $\mathcal{P}_1$ *and* $\mathcal{P}_2$ *decreases distances by at most factor of* 2*, that is,* $||\boldsymbol{\mu}_i' - \boldsymbol{\mu}_j'||_2 \geq \frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{2}$*, and* $||\boldsymbol{\mu}_i'' - \boldsymbol{\mu}_j''||_2 \geq \frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{2}$*.*

Now we show that if the slope ratio is at least $10\sqrt{\log k}$ then the partition where each coordinate is picked with probability 1/2 is good with probability at least $1-\frac{1}{k^2}$.

Lemma 3.1 *Let* $h_{\min} = \min_{i,j}\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty}$*. If* $h_{\min} \geq 10\sqrt{\log k}$ *then with probability at least* $1-\frac{1}{k^2}$*, a random partition* $(\mathcal{P}_1, \mathcal{P}_2)$ *is a good partition.*

The proof is a simple consequence of Chernoff bounds.

Analysis: Correct clustering of $\mathcal{S}_0$. Suppose the algorithm is at the step where $\tilde{\mathcal{C}}_i = \mathcal{S}_0 \cap \mathcal{C}_i$; we show that if the coordinate partition is good, then with high probability the algorithm will terminate at this point.

In what follows, to avoid confusing notation, we will assume that all the data is already projected down to either $\mathcal{P}_1$ or $\mathcal{P}_2$, and the goal is to show that if the separation of the centers (in the projected space) exceeds the minimum threshold, then the probability of error is at most δ.

We begin with a simple lemma stating that median points of $\tilde{\mathcal{C}}_i$ will be close to actual distribution centers with high probability.

Lemma 3.2 *If sample* $\mathcal{S}_0$ *has size at least* $\frac{96dk\log\frac{dk}{8\delta}}{w_{\min}}$*, then with probability at least* $1-\delta/2$*, the following holds:*

$$||\tilde{\boldsymbol{\mu}}_i - \boldsymbol{\mu}_i||_\infty \leq \frac{R}{\sqrt{dk}} \tag{5}$$

Moreover, for $\boldsymbol{s} \in \boldsymbol{D}_i$*, and each coordinate* j*,*

$$\Pr\left[s_j \in [\mu_{ij}, \tilde{\mu}_{ij}]\right] \leq \frac{1}{4\sqrt{dk}}, \tag{6}$$

where $\tilde{\boldsymbol{\mu}}_i = \operatorname{med}(\mathcal{S}_0 \cap \mathcal{C}_i)$*, and* R *is the* $\frac{1}{2}$*-radius of the underlying distribution on each coordinate.*

Now we are ready to state the main technical lemma of this part, which asserts that given slightly perturbed centers, all but a small fraction of points will be classified correctly.

Lemma 3.3 *Suppose the distribution centers satisfy the separation condition, and a set of points $\{\tilde{\boldsymbol{\mu}}_1, \tilde{\boldsymbol{\mu}}_2, \ldots \tilde{\boldsymbol{\mu}}_k\}$ satisfies* (5) *and* (6) *for all i. Then for any $\boldsymbol{s}$ drawn from $\boldsymbol{D}_i$, with probability at least $1-\varepsilon$, we have $||\boldsymbol{s} - \tilde{\boldsymbol{\mu}}_i||_1 \leq ||\boldsymbol{s} - \tilde{\boldsymbol{\mu}}_j||_1$.*

We summarize the results of this part in the following theorem.

Theorem 3.4 *Suppose $\mathcal{P} = (\mathcal{P}_1, \mathcal{P}_2)$ is a good partition, suppose sample $\mathcal{S}_0$ has size at least $\Theta(\frac{d \log \frac{dk}{\delta\varepsilon}}{w_{\min}})$, and let $\tilde{\boldsymbol{\mu}}_i = \operatorname{med} \mathcal{S}_0 \cap \mathcal{C}_i$. Then with probability at least $(1-\delta)$ the following holds. For any sample x drawn from any of the $\boldsymbol{D}_i$, the probability of x being misclassified is at most ε.*

Proof. From lemma 3.2, it follows that the conditions of lemma 3.3 are satisfied with probability at least $(1-\delta)$, and the result follows. ■

Analysis: Stopping condition. We show that the algorithm stops when both $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are very close to the true clustering. The proof is based on the following idea. For random $\mathbf{x} \in \boldsymbol{D}_i$, the assignments in $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are fully independent from each other. Let p'_{ij} and p''_{ij} denote the probabilities that a random sample from $\boldsymbol{D}_i$ will be assigned to $\tilde{\mathcal{C}}'_j$ and $\tilde{\mathcal{C}}''_j$ respectively. Recall that the algorithm stops only if $Y = \sum_i \tilde{\mathcal{C}}'_i \triangle \tilde{\mathcal{C}}''_i$, is small. We prove correctness of the algorithm by showing that (i) Y is tightly concentrated around its expectation and (ii) the expectation is small only if clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are approximately correct.

We start with the expectation of Y. Obviously the expected contribution of $\mathbf{x} \in \boldsymbol{D}_i$ to $\tilde{\mathcal{C}}'_j \triangle \tilde{\mathcal{C}}''_j$ is $p'_{ij}(1-p''_{ij}) + p''_{ij}(1-p'_{ij})$, and thus

$$\begin{aligned}\mathbf{E}[Y] &= \textstyle\sum_i |\mathcal{C}_i| \sum_j (p'_{ij}(1-p''_{ij}) + p''_{ij}(1-p'_{ij})) \geq \\ &\geq m w_{\min} \textstyle\sum_i \sum_j (p'_{ij}(1-p''_{ij}) + p''_{ij}(1-p'_{ij})).\end{aligned}$$

Therefore $\mathbf{E}[Y]$ is only small when for i and j p'_{ij} and p''_{ij} are close to each other, and furthermore each is close to 0 or 1. To show that Y is concentrated around its expectation, we prove the following lemma.

Lemma 3.5 *For any set of probabilities $\{p'_{ij}\}$ and $\{p''_{ij}\}$, we have* $\Pr[|Y - \mathbf{E}[Y]| > \max(\mathbf{E}[Y]/2, \varepsilon m)] < \exp(-\varepsilon m/12)$.

Proof. Each sample $x \in \boldsymbol{D}_i$ is independent and contributes 1 to Y with fixed probability. Thus we can use Chernoff bounds to show that Y is concentrated. We use the following version of the bound:[4] $Pr[|Y - \mathbf{E}[Y]| > t|] < \exp\left(\frac{-t^2}{4(t+\mathbf{E}[Y])}\right)$. Choosing $t = \max(\mathbf{E}[Y]/2, \varepsilon m)$, we have the required probability. ■

Now we prove the main lemma of this section, which says that with probability $1 - \exp[-\frac{\varepsilon m}{20}]$ the algorithm stops only after encountering an approximately correct clustering.

Lemma 3.6 *With probability $1 - \exp[-\frac{\varepsilon m}{20}]$, the stopping criteria accepts a pair of clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ only if there exists a matching π such that $\sum \tilde{\mathcal{C}}'_i \triangle \mathcal{C}_{\pi_i} \leq \varepsilon m$.*

Proof. Note that there are fewer than $(k+1)^{|S_0|}$ possible clusterings of the training sample, and each gives rise to possibly different sets of assignment probabilities p' and p''. By lemma 3.5 and the union bound, the probability that for *all* partitions, Y is below $\max(2\mathbf{E}[Y], \varepsilon m)$ is bounded by $1-(k+1)^{|S_0|}\exp(-\varepsilon m/10) \leq 1-\exp(-\varepsilon m/20)$, where we have used that $m \geq \frac{20|S_0|\log k}{\varepsilon}$. In other words with high probability, for each clustering of S_0, the expected value of the symmetric difference on S_1 is not too far from the actual value.

Now consider the point where the algorithm stops; the two clusterings $\tilde{\mathcal{C}}'$ and $\tilde{\mathcal{C}}''$ are such that $Y \leq \varepsilon m$ and thus $\mathbf{E}[Y] \leq Y + \varepsilon m \leq 2\varepsilon m$. But

$$\mathbf{E}[Y] \geq w_{\min} m \sum_i \sum_j (p'_{ij}(1-p''_{ij}) + p''_{ij}(1-p'_{ij})),$$

and since all terms are non-negative for any i and j we have $p'_{ij}(1-p''_{ij}) + p''_{ij}(1-p'_{ij}) \leq \frac{2\varepsilon}{w_{\min}}$. Obviously this holds only if $p''_{ij}, p'_{ij} \notin [\frac{4\varepsilon}{w_{\min}}, 1 - \frac{4\varepsilon}{w_{\min}}]$ and $|p'_{ij} - p''_{ij}| < 0.5$. Finally, recall that the stopping condition ensures that each cluster in $\tilde{\mathcal{C}}'$ is large, and hence the matrix $\{p'_{ij}\}$ is indeed close to a permutation matrix. Thus there is a permutation π such that for any point $x \in \boldsymbol{D}_i$, the probability of misclassification is at most $\frac{4\varepsilon}{w_{\min}}$; applying standard tail inequalities we immediately have the desired result. ■

Finally, by unifying the results of the previous sections, we can formulate the proof of the main theorem.

Theorem 3.7 *Suppose a mixture of k distributions $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_k$, in $\Re^d$, is such that either one of the following conditions is satisfied:*

- *For every i and j, $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega\left(R\sqrt{\frac{k}{\varepsilon}}\right)$ and $\frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty} \geq \Omega\left(\sqrt{\frac{k}{\varepsilon}}\right)$ or*

4 One can verify it by replacing the denominator with $\max(t, \mathbf{E}[Y])$.

- *Each distribution belongs to class* $\mathcal{F}_1$ and *for every* i *and* j, $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2 \geq \Omega\left(\frac{Rk^{5/2}\sqrt{\log k}}{\varepsilon^2}\right)$.

Then, given a sample $\mathcal{S}$ *of size at least* $\tilde{\Omega}(\frac{dk}{w_{\min}})$, *with probability at least* $1-\delta$ *the algorithm classifies all the samples correctly, except for at most an* $\varepsilon/w_{\min}$ *fraction of them. The time taken is due to the exhaustive clustering of* $\mathcal{S}_0$ *and hence is exponential in both* d *and* k.

Proof. We present the proof of the algorithm when the condition $h_{\min} > 10\sqrt{\log k}$ is satisfied i.e. the centers are not aligned along a few axes. In particular, this includes the case when all pairs of distributions satisfy the first bulleted condition above. The general case follows immediately from the preprocessing step, which is detailed in the appendix. The algorithm proceeds by first choosing the set $\mathcal{S}_0$ and then partitioning the coordinate set into $(\mathcal{P}_1, \mathcal{P}_2)$. Hence, putting together the result from Theorem 3.4 and Lemma 3.6, it follows that, in one iteration of steps $\mathbf{3}a-3e$:

1. Partition $(\mathcal{P}_1, \mathcal{P}_2)$ is balanced with probability at least $1 - \frac{1}{k^2}$.
2. If the clustering $\{\tilde{\mathcal{C}}_1, \ldots, \tilde{\mathcal{C}}_k\}$ of $\mathcal{S}_0$ matches the actual clustering $\{\mathcal{C}_1, \ldots, \mathcal{C}_k\}$ then the centers $\{\tilde{\boldsymbol{\mu}}_i\}$ induce an ε-error clustering with probability at least $1-\delta$.
3. The stopping criterion accepts only an $\varepsilon/w_{\min}$-error clustering with probability $1 - \exp[-\frac{\varepsilon m}{20}]$.

Then, conditioned on the fact that $h_{\min} > 10\sqrt{\log k}$, the union bound on the above error probabilities implies that the total probability of getting an $\varepsilon/w_{\min}$-error clustering in $\log\left(\frac{1}{\delta}\right)$ iterations is at least $1 - \left(1 - \frac{1}{k^2} - \delta\right)^{\log\left(\frac{1}{\delta}\right)} - \log\left(\frac{1}{\delta}\right)\exp[-\frac{\varepsilon m}{20}] \geq 1 - 2\delta$. ■

3.2. Large separation between centers

In this part we show that if the distances between centers is at least at least $\Omega(R\sqrt{d})$, then performing exhaustive search on a set S_0 of size only $\Theta(\frac{\log dk}{w_{\min}})$ would suffice. This immediately results in an algorithm that is polynomial in the number of dimensions. Note that this result is similar in strength to Dasgupta's original algorithm [6], yet allowing a much wider class of distributions.

Theorem 3.8 *Suppose a mixture of* k *distributions* $\boldsymbol{D}_1, \ldots, \boldsymbol{D}_k$, *in* $\Re^d$, *satisfy condition of Theorem 3.7, and in addition the distance between centers is at least* $R\sqrt{d}$. *Then, given a sufficiently large sample, we can provide the same guarantees as in Theorem 3.7.*

We again omit the proof of this theorem, as it is almost same as that of Theorem 3.7. The only difference a is that given separation $\Omega(R\sqrt{d})$, we can allow the approximated center to be within $\Theta(R\sqrt{d})$, of the true center. The polynomial time bound follows as the number of clusterings that are computed from the test set $\mathcal{S}_0$ is $(k+1)^{|\mathcal{S}_0|}$ and hence is polynomial in d.

Note that because of the stopping condition, our algorithm will never produce an invalid clustering. Thus one can make the algorithm run in close to the minimum possible time, by simply starting with just k samples and doubling it every time the algorithm finds no suitable clustering.

4. Minimal separation results

In order to obtain a lower bound showing that a certain separation between centers is in fact necessary to learn mixtures, we consider the *Bayes-optimal* method for classification: given exact knowledge of two density functions ρ_1 and ρ_2, with equal mixing weights, one should classify a point x as coming from the first distribution if $\rho_1(x) > \rho_2(x)$. This classification is incorrect with probability $\rho_2(x)/(\rho_1(x)+\rho_2(x))$. No method can achieve a misclassification probability better than this. Note that since the Bayes-optimal algorithm has no notion of scale, it suffices to consider $R = 1$.

In this section we show that for a fixed number of components k, and a fixed accuracy, a separation of at least $\Omega(R)$ is necessary, even for the Bayes-optimal algorithm. It is enough to assume $k = 2$ and equal mixing weights. We start with the following simple observation: if the Bayes-optimal algorithm encounters points on which it is not confident (i.e. the ratio of probability densities is close to 1) with constant probability, then it classifies points incorrectly with constant probability.

Lemma 4.1 *Suppose* $c > 0$, *and we have a mixture of distributions* $\mathcal{D}_1$ *and* $\mathcal{D}_2$ *with density functions* $\rho_1(x)$ *and* $\rho_2(x)$. *If for* $x \in \mathcal{D}_1$, *the ratio of densities* $\rho_1(x)/\rho_2(x) \leq c$ *with probability at least* t. *then in the case of equal mixing weights, the Bayes-optimal algorithm will make an error with probability at least* $\frac{t}{2(c+1)}$.

Proof. Indeed, consider the set $M = \{x \in \Re^d | \rho_1(x)/\rho_2(x) \leq c\}$. Let $\rho_1(M)$ and $\rho_2(M)$ be probability densities concentrated in M, according to $\mathcal{D}_1$ and $\mathcal{D}_2$. Obviously $\rho_2(S) \geq \frac{\rho_1(M)}{c} \geq t$. Now consider the set $M_1 = \{x \in M | \rho_1(x) \geq \rho_2(x)\}$. The Bayes-optimal algorithm will assign a point in M_1 to the first distribution, and a point in $M_2 = M - M_1$ to the second.

Obviously, if $\rho_1(M_2) \geq \frac{t}{c+1}$, then the optimal algorithm would make a mistake with probability at least $\frac{t}{c+1}$. Otherwise, if $\rho_1(S_2) \leq \frac{t}{c+1}$, then $\rho_1(M_1) \geq t - \rho_1(M_2) \geq \frac{ct}{(c+1)}$, and thus $\rho_2(M_1) \geq \frac{t}{c+1}$, since $M_1 \subseteq M$. Since the distributions have equal weight, the lemma follows immediately. ■

We now show that, for certain classes of distributions, if the minimum separation is less than an absolute constant c_1, then the misclassification error is at least an absolute constant c_2. This establishes a sense in which a constant separation on centers is asymptotically necessary in at least some cases.

Lemma 4.2 *There exist constants c_1 and c_2, independent of n, so that the following holds. For a mixture of two Cauchy distributions $\mathcal{D}_1(0,1)$ and $\mathcal{D}_2(\boldsymbol{\mu},1)$, with equal mixing weights, if $||\boldsymbol{\mu}||_2 < c_1$, then the Bayes-optimal algorithm will misclassify a random sample from $\mathcal{D}_1$ with probability at least c_2.*

Proof. We take $c_1 = 1/2$. It is enough to show that for a random point drawn from $\mathcal{D}_1$, the following holds with probability at least 2/3: $t(x) = \frac{\rho_1(x)}{\rho_2(x)} \leq 8$ where $\rho_1(x) = \prod \frac{\pi/2}{x_i^2+1}$ and $\rho_2(x) = \prod \frac{\pi/2}{(x_i-\mu_i)^2+1}$ are the probability densities of $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively.

We show this by estimating $\ln t(x)$. We have $\ln t(x) = \ln \frac{\rho_1(x)}{\rho_2(x)} = \sum \ln\left(1 + \frac{-2\mu_i x_i + \mu_i^2}{x_i^2+1}\right)$, and using $\ln(1+x) \leq x$, we have

$$\ln t(x) \leq \sum \frac{-2\mu_i x_i + \mu_i^2}{x_i^2+1} \leq \left|\sum \frac{-2\mu_i x_i}{x_i^2+1}\right| + ||\boldsymbol{\mu}||_2^2. \tag{7}$$

The second term is at most $c_1^2 = \frac{1}{4}$, so we just need to upper-bound the first sum. Recall that x is drawn from $\mathcal{D}_1(0,1)$, and so for any i, we have $\mathbf{E}\left[\frac{\mu_i x_i}{x_i^2+1}\right] = 0$, and $\sigma^2\left(\frac{2\mu_i x_i}{x_i^2+1}\right) \leq 4\mu_i^2$, where the former follows from the symmetry of x_i around 0, and the latter from the fact that $\frac{x_i}{x_i^2+1} < 1$. Thus for x drawn from $\mathcal{D}_1$, we have $\Pr\left[\left|\sum \frac{-2\mu_i x_i}{x_i^2+1}\right| \geq 3.5||\boldsymbol{\mu}||_2\right] \leq 1/3$. Combining this with (7) we have $\Pr\left[\ln t(x) \leq \frac{7}{4} + \frac{1}{4} \leq 2\right] > 2/3$, and hence $t(x) \leq 8$ with probability at least 2/3. ■

Note, that exactly the same proof generalizes to arbitrary power laws. For fixed power coefficient the ratio between necessary and sufficient conditions will be a constant. For exponential distributions, it is easy to see that our algorithm from Section 2, with known centers, is Bayes-optimal.[5] By rewording the proof of the lemma 2.1, we can obtain necessary condition as well. Finally, for Gaussian distributions, the result follows from the fact that choosing the L_2 closest center is Bayes-optimal, and thus the separation of at least $\Omega(\sigma) = \Omega(R)$ is necessary.

5 This can be seen by taking logarithms of density functions, and noting that ln is monotonic

5. Conclusions and Open Problems

We have presented a new technique for learning arbitrary mixtures of distributions using the L_1 norm. Through probabilistic analysis, we were able to show that a very simple algorithm can correctly learn almost arbitrary distributions.

Now we outline open problems with respect to learning mixture models. Perhaps the most intriguing question is existence of a polynomial time algorithm. One potential way to try achieving this goal is to replace exhaustive search step by some variation of the k-median problem defined on the set $\mathcal{S}_0$. Although the general k-median problem is NP-hard, it is plausible that in the special case where the input comes from a mixture of distributions, there might exist a polynomial-time algorithm that provides strong enough guarantees for our purposes, with high probability.

Another approach would be the projection of the sample to a lower-dimensional subspaces while preserving the cluster structure[15, 6, 1]. However, it can be shown that orthogonal projections are not very useful with infinite variances, leaving open the question of finding an appropriate projection.

We have shown that separation $\Theta(R)$ is needed to learn mixtures of power-law distributions and Gaussian distributions; however the constant changes for different distributions. Can this proof be generalized to have a single constant for all distributions? A final issue is to produce logarithmic dependence on the error-rate; this would be a very interesting extension of our work, as it would show that even a slight increase in separation will increase the learning rate dramatically. As a first preliminary result in this direction, we can show that such bounds are indeed possible at the expense of an additional constraint on the slope.

References

[1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *COLT*, 2005.

[2] S. Arora and R. Kannan. Learning mixtures of arbitrary gaussians. In *ACM Symposium on Theory of Computing*, pages 247–257, 2001.

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proc. STOC*, 2001.

[4] V. Barnett and T. Lewis. *Outliers in statistical data.* John Wiley & Sons, 1994.

[5] T. Cover and P. Hart. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, 13(21-27), 1967.

[6] S. Dasgupta. Learning mixtures of gaussians. In *FOCS99*, 1999.

[7] S. Dasgupta and L. Schulman. A two-round variant of em for gaussian mixtures. In *Sixteenth Conference on Uncertainty in Artificial Intelligence (UAI)*, 2000.

[8] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood estimation from incomplete data via the em algorithm (with discussion). *Royal Statistical Soc. B*, 39:1–38, 1977.

[9] Y. Dodge, editor. *L_1-statistical analysis and related methods.* Elsevier Science Publishers B.V., 1992.

[10] Y. Dodge, editor. *Statistical data analysis based on the L_1 norm and related methods.* Elsevier Science Publishers B.V., 2002.

[11] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for mixture models. In *ECCC*, 2004.

[12] B. G. Lindsay. Mixture models: Theory, geometry and applications. *NSF-CBMS Regional Conference Series in Probability and Statistics IMS/ASA*, 5, 1995.

[13] P. Rousseuw and A. Lerow. *Robust Regression and Outlier Detection.* Wiley-Interscience, 2003.

[14] D. Titterington, A. Smith, and U. Makov. *Statistical Analysis of Finite Mixture Distributions.* Wiley, 1985.

[15] S. Vempala and G. Wang. A spectral algorithm for learning mixture models. *J. Comput. Syst. Sci.*, 68(4):841–860, 2004.

A. Aligned Centers: Preprocessing Step

Recall that Algorithm 1 works if in addition to the separation condition, the slope for each pair is at least $\Omega(\sqrt{\log k})$. Therefore if the algorithm fails, there will be pairs of centers i and j, such that $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty \geq \frac{||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_2}{\sqrt{\log k}}$. The preprocessing step will pre-cluster the data in such a way that each part will now only contain distributions that satisfy slope condition. The algorithm works as follows. We look for a coordinate i for which there is at least a pair of centers, $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$ say, such that the distance between μ_{1i} and μ_{2i} is large. This means that the projections of all the sample data on this coordinate will form at least two intervals which are separated by a sparse interval having at most an $\frac{\varepsilon}{k}$ fraction of the points. We will split the sample into *superclusters* that are defined by these intervals on the i^{th} coordinate.

Without loss of generality we assume that $w_{\min} \geq \varepsilon$.

Algorithm 2

Input: Sample $\mathcal{S}$, and coordinate i,

Output: A collection of k' superclusters,

Description:

1. *Check that $\mathcal{S}$ has at least $w_{\min}m/2$ samples; else return $\mathcal{S}$ as a supercluster.*
2. *Look at the i^{th} coordinates of all the sample points. Find the lower and upper limits B_l and B_r such that the interval $[B_l, B_r]$ has the middle $1 - \frac{\varepsilon}{2k}$ fraction of the points. Delete all samples whose coordinates fall outside this range.*
3. *Partition $[B_l, B_r]$ into intervals of size $s = \max\left(\frac{B_r - B_l}{10k}, \frac{10Rk}{w_{\min}\varepsilon}\right)$, with the h^{th} interval being $I_h = [B_l + (h-1)s, B_l + hs]$.*
4. *From the list $\{I_h\}$, find an interval $I = [I_l, I_r]$ such that I has at most $\frac{\varepsilon m}{k}$ samples, and there are at least $w_{\min}m/2$ samples on each side of I. If no such interval can be found then* **return** *$\mathcal{S}$ as a single supercluster.*
5. *Partition the sample set as $\mathcal{S}_1 = \{\boldsymbol{s} \in \mathcal{S} | s_i < I_l\}$ and $\mathcal{S}_2 = \{\boldsymbol{s} \in \mathcal{S} | s_i > I_r\}$. Delete all the samples whose i^{th} coordinate falls in I.*
6. *If the value of s was chosen to be $\frac{10Rk}{w_{\min}\varepsilon}$, then return the parts $\mathcal{S}_1$ and $\mathcal{S}_2$. Else call the algorithm recursively with $(\mathcal{S}_1, \{i\})$ and $(\mathcal{S}_2, \{i\})$.*

Once we cluster our data, we essentially run the original algorithm on each supercluster, while guessing the number of centers that lie in this supercluster. After we have computed the k possible centers, they are again tested out in the cross-validation phase as before. The following lemma summarizes the effects of running the preprocessing algorithm on the data; the proof is given in the full version of the paper.

Lemma A.1 *Given $m > \frac{10k\log(\frac{d}{\delta})}{\varepsilon w_{\min}}$ samples, the preprocessing satisfies the following condition with probability at least $1 - \delta$.*

1. *If $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_j$ are such that $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_j||_\infty > \frac{300Rk^2}{\varepsilon^2}$, then samples from $\boldsymbol{D}_i$ and $\boldsymbol{D}_j$ will be in separate superclusters, except for at most an ε-fraction of points.*
2. *If centers $\boldsymbol{\mu}_i$ and $\boldsymbol{\mu}_{i'}$ are in the same supercluster, then after deleting the preprocessed coordinates, the distance $||\boldsymbol{\mu}_i - \boldsymbol{\mu}_{i'}||_2$ does not decrease by more than $\frac{100k^{5/2}R}{\varepsilon^2}$ additively.*
3. *For each sample $\mathbf{x}$ drawn from the i^{th} distribution $\boldsymbol{D}_i$, with probability $1 - \varepsilon$, $\mathbf{x}$ will ultimately be classified in the same supercluster as $\boldsymbol{\mu}_i$.*