Disentangling Gaussians

Adam Tauman Kalai*
Microsoft Research, New England
Cambridge, MA 02139
adum@microsoft.com

Ankur Moitra†
Institute for Advanced Study
Princeton, NJ 08540
moitra@ias.edu

Gregory Valiant‡
University of California, Berkeley
Berkeley, CA 94720
gvaliant@eecs.berkeley.edu

1. INTRODUCTION

The Gaussian mixture model (GMM) is one of the oldest and most widely-used statistical models. It is comprised of a weighted combination of heterogeneous Gaussian sources. As a simple one–dimensional example, consider measurements of heights of adults in a certain population, where the distribution of heights can be closely approximated as a mixture of two univariate Gaussians, one for males and one for females [3]. Can one recover the parameters of the Gaussians from unlabeled height measurements alone (with no gender information)?

This paper focuses on the case where the mixture consists of a small but unknown number of Gaussians that may *overlap*—the combined density may even have a single peak, as in the height example, and the dimensionality may be high. Much of the previous work on this problem attempts to learn the parameters through clustering, and consequently needs to make a strong separation assumption on the components in the mixture. The primary contribution of our research is to avoid this assumption by instead basing our learning algorithm upon the algebraic structure of the mixture. Our algorithm succeeds even if the components overlap almost entirely, in which case accurate clustering is no longer possible. We give a simple notion of "condition number" of a GMM which characterizes its complexity up to polynomial factors. Generally speaking, the conclusion is that the *statistical complexity* and *computational complexity* of this general problem is in every way polynomial except for the dependence on the number of Gaussians, which is necessarily exponential.

Statisticians have long known that from random samples from a GMM it is possible to identify the Gaussians *in the limit*—one can eventually recover to arbitrary precision each subpopulation's mean, variance, and proportion, given sufficiently many examples [14]. However, their analysis provides no bounds on convergence rate—it might be *exponentially* slow even for two Gaussians in one dimension. Moreover, heuristics in widespread use, such as the EM algorithm, suffer from local minima and have weak theoretical guarantees.

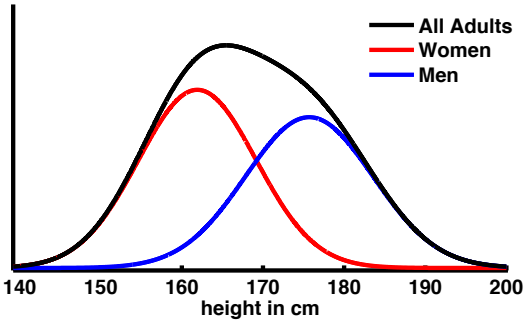

Figure 1: The Gaussian approximations of the heights of adult women (red) and men (blue). Can one recover estimates of these Gaussian components given only the aggregate data without gender labels (black)? [Raw data from National Health and Nutrition Examination Survey (NHANES)]

In seminal work, Dasgupta [5] put forth a polynomial-time *clustering* approach for learning GMMs that is provably accurate under certain assumptions. In particular, if the Gaussians are sufficiently "well separated" then one can usually group together all the samples originating from the same Gaussian. As a by-product, parameter estimation follows easily by estimating the mean, covariance matrix, and proportion of samples from each cluster separately. In rapid succession, increasingly powerful clustering techniques that require "separability assumptions" between all pairs of Gaussians have since been analyzed [2, 6, 16, 11, 1, 4] .

In some sense, the parameter estimation problem is more fundamental than clustering, because accurate parameter estimates can be easily used for accurate clustering. And in the general case where the Gaussians may overlap, it is impossible to cluster the points, with any degree of confidence, just as one cannot accurately predict gender from the fact that a person is, say, 1.7 meters tall. Nonetheless, from height data alone one may still recover the means, variances, and mixing weights of the two constituent Gaussians. Hence, while one cannot hope to cluster the individual samples, one can decompose the Gaussian mixture and efficiently disentangle the Gaussian sources.

This work appeared as "Efficiently Learning Mixtures of Two Gaussians" (Kalai, Moitra, Valiant, STOC 2010) and "Settling the Polynomial Learnability of Mixtures of Gaussians" (Moitra, Valiant, FOCS 2010).

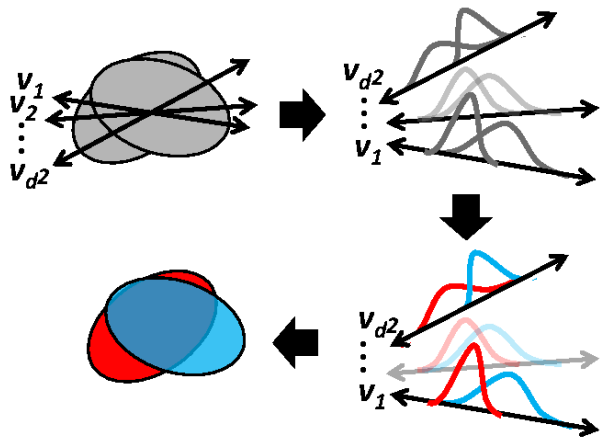

Figure 2: Illustration of the high–level approach: 1. project the data onto a series of vectors and learn the parameters of the resulting one dimensional GMMs, 2. determine a consistent labeling between the components of the recovered one dimensional GMMs, and 3. for each component, combine the recovered one dimensional parameters to reconstruct an estimate of the high dimensional parameters.

1.1 Our approach

We describe a polynomial-time GMM learning algorithm—we emphasize that throughout, we favor clarity of presentation and ease of analysis at the cost of impracticality. The algorithm is based on the random projection method (see, e.g., [15]). Since the projection of a multinormal distribution onto one dimension is a normal distribution, the projection of a GMM is a GMM in one dimension. Roughly, the algorithm proceeds by projecting the data onto a sequence of vectors, solving the associated sequence of one–dimensional problems, and then reconstructing the solution to the original high–dimensional GMM problem.

There are several clear obstacles that must be surmounted to consummate this approach. First and foremost, is the question of solving the problem in one dimension. one–dimensional problems are often easy if one is willing to resort to brute force search; for the problem of learning GMMs, it has already been mentioned that the bounds are necessarily exponential in k, hence one might expect that a crude brute-force search would suffice to solve the one–dimensional case. We show that this is true. The difficulty, of the one–dimensional case is not in designing the algorithm, but in guaranteeing that this crude brute-force search can be conducted over a sufficiently coarse grid. We show this by establishing, what we term, the *polynomially robust* identifiability of GMMs, which we discuss in Section 4.

Supposing one has an efficient algorithm for the one dimensional problem, a second obstacle in our high–level approach is ensuring that the projected data that are given as inputs to the one–dimensional algorithm are meaningful. Consider, for example, a GMM that consists of two Gaussian components that have identical covariances, but different means. If, unluckily, we project the data onto a vector orthogonal to the difference in the means, then the resulting one–dimensional mixture will have just a single component. Further complicating this concern is the existence of GMMs, such as that depicted in Figure 3, for which two or more essentially non-overlapping components will, with very high probability, project to nearly identical Gaussians in a random projection. How can we hope to disentangle these components if, in nearly every projection, they are indistinguishable?

We demonstrate that this problem can only arise if a certain spectral condition holds. In fact, if this spectral condition is met we will be able to take another, orthogonal approach to learning – namely, *clustering*. Consider a partition of the components in the original mixture. This partition in turn generates two sub-mixtures, each of which is a GMM which inherits components of the original mixture corresponding to one of the two sets in the partition. When our spectral condition is met, we will be able to cluster the samples into two groups so that (for some partition of the components in the mixture), each group of samples was generated by taking the appropriate number of samples from one of the two sub-mixtures. We can then apply our algorithm recursively to each of the two clusters separately, to learn both sub-mixtures.

For clarity of exposition, in Section 3 we first describe a simplified version of our algorithm that does not require the clustering and recursion steps, though has slightly weaker performance guarantees. In Section 5, we describe the full algorithm.

2. MODEL AND RESULTS

The input to our algorithm is a sample set of n points in d dimensions, drawn independently from GMM $F = \sum_{i=1}^{k} w_i F_i$, which has *mixing weights* $w_i > 0$ and $\sum_i w_i = 1$, and k different *Gaussian sources* F_i, where each $F_i = \mathcal{N}(\mu_i, \Sigma_i)$ is a *distinct* d-dimensional Gaussian with mean $\mu_i \in \mathbf{R}^d$ and covariance matrix $\Sigma_i \in \mathbf{R}^{d\times d}$. The output is an estimate GMM $\hat{F} = \sum_{i=1}^{\hat{k}} \hat{w}_i \hat{F}_i$. The goal of the estimation algorithm is to correctly learn k and furthermore to approximate the parameter *set* $\{(w_1, \mu_1, \Sigma_1), \ldots, (w_k, \mu_k, \Sigma_k)\}$. Note that one cannot hope to learn the correct ordering of the components F_i, since any permutation results in an identical distribution.

To measure the distance between Gaussians $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$, we employ the *statistical distance*. This natural metric is defined, more generally, for probability distributions. For distributions G and G' with respective probability density functions g and g', the statistical distance is defined as

$$D_{tv}(G, G') = \frac{1}{2}\int_{\mathbf{R}^d} |g(x) - g'(x)|dx \in [0, 1].$$

This quantity is zero if the distributions are identical and one if the distributions have disjoint support. The main strength of statistical distance as a metric is that it measures the information theoretic similarity of two distributions; for instance, if two distributions A and B have $D_{tv}(A, B) = .01$, then *no* algorithm can distinguish a set of 100 samples from A from a set of 100 samples from B with probability more than .6. A second advantage of statistical distance is that it is scale invariant and affine invariant, meaning that an identical rescaling of two distributions does not affect the distance. In contrast, Euclidean error estimates such as $\|\mu - \mu'\|_2$ and $\|\Sigma - \Sigma'\|_F$ change as one rescales the problem.

Moreover, guaranteeing low statistical distance implies low

We now define a quantity that controls "how hard" it is to learn a particular GMM.

DEFINITION 1. *The condition number $\kappa(F)$ of GMM $F = \sum_{i=1}^{k} w_i F_i$ is defined to be,*

$$\kappa(F) = \min\big(\{w_1, w_2, \ldots, w_k\} \cup \{D_{tv}(F_i, F_j) \mid i \neq j\}\big).$$

It is not hard to see that any estimation algorithm requires, at a minimum, a number of samples inversely proportional to $\kappa(F)$ to have any chance of accurate estimation. The reason is that one requires $1/w_i$ samples to have a constant probability of encountering a single sample generated by F_i. Hence, for very small w_i, a large sample size is necessary. Similarly, even if one exactly knows two probability distributions F and F', one requires $1/D_{tv}(F, F')$ samples to have a constant probability of distinguishing between the case that all samples arise from F or all samples arise from F'. Hence, an inverse dependence on $\kappa(F)$ is required, while our bound is polynomial in $1/\kappa(F)$.

We are now ready to state our main theorem.

THEOREM 1. *For every $k \geq 1$, there is a constant $c_k > 0$ dependent on k such that the following holds. For any d-dimensional GMM $F = \sum_{i=1}^{k} w_i F_i$, $\epsilon, \delta > 0$, and $n > \left(\frac{d}{\epsilon\delta\kappa(F)}\right)^{c_k}$, the estimation algorithm run on $x_1, x_2, \ldots, x_n$ independently drawn from F outputs GMM $\hat{F} = \sum_{i=1}^{\hat{k}} \hat{w}_i \hat{F}_i$ such that, with probability $\geq 1 - \delta$, $\hat{k} = k$ and there exists a permutation π on $\{1, 2, \ldots, k\}$ such that*

$$|w_i - \hat{w}_{\pi(i)}| \leq \epsilon \text{ and } D_{tv}(F_i, \hat{F}_{\pi(i)}) \leq \epsilon \text{ for } i = 1, 2, \ldots, k.$$

The runtime of the estimation algorithm is polynomial in the number of samples, n.

2.1 Interpretation

Our algorithm takes as input samples from a d-dimensional GMM F, and outputs estimates of the parameters of F. The main theorem shows that as the number of samples, n, increases the error of our estimates decreases as $O(\left(\frac{d}{n}\right)^{\alpha})$, where α is a constant that only depends on the number of sources, k. This rate of convergence is striking both because it exhibits a polynomial (as opposed to exponential) dependence on the number of dimensions, d and because it exhibits an inverse polynomial dependence on the number of samples. This is in contrast to the inverse logarithmic rate of convergence that has been hypothesized [9]. Lastly, the big-Oh notation hides a leading constant that exhibits an inverse polynomial dependence on the condition number of F, $\kappa(F) > 0$.

In applications in which a learner has a reasonable upper bound on k and lower bound on $\kappa(F)$, we could determine upper bounds on how much data is required to learn a good estimate – and run our algorithm accordingly. In contrast if either an upper bound on k or a lower bound on $\kappa(F)$ is missing, *every* estimator can fooled into either outputting a mixture with the wrong number of components, or outputting a mixture which is far from the true GMM.

While the runtime and data requirement of our algorithm scale super–exponentially as a function of the number of Gaussian components, we show that, unfortunately, an exponential dependence on k is necessary at least in the case where the Gaussians overlap significantly. We give a simple construction of a mixture of k (polynomially) overlapping Gaussians that are exponentially close to a single Gaussian. This state of affairs is in contrast to the aforementioned clustering algorithms which depend only polynomially on k. Hence, when there are a large number of very well-separated Gaussians, clustering seems to be a better approach.

2.2 History and related work

Perhaps the earliest GMM study was conducted by Karl Pearson in the 1890's [12]. He was given a dataset consisting of measurements of 1000 crabs found near Naples, and conjectured that the dataset arose as a GMM of two components, corresponding to two crab species. Pearson then attempted to recover estimates of the parameters of the two hypothesized species, using the *method of moments*. In particular, he computed empirical estimates of the first six (raw) moments $\mathrm{E}[x^i] \approx \frac{1}{m}\sum_{j=1}^{m} x_j^i$, for $i = 1, 2, \ldots, 6$ from sample points $x_1, \ldots, x_m \in \mathbf{R}$. Using only the first five moments, he solved a cleverly constructed ninth-degree polynomial, *by hand*, from which he derived a set of candidate mixture parameters. Finally, he heuristically chose the candidate among them whose sixth moment most closely agreed with the empirical estimate.

The potential problem with this approach, which Pearson acknowledged, was the issue of robust identifiability. Perhaps there exist two different mixtures, where the components of one mixture are very different from the components of the other mixture, but nevertheless the densities and the moments of the two mixtures are extremely similar. We show that this cannot be the case, in some sense validating Pearson's approach.

Recently, propitiated by the emergence of huge, high–dimensional datasets, the question of learning GMMs was revisited by the Theoretical Computer Science community. In this body of work, initiated by Dasgupta [5], a line of computer scientists designed *polynomial time* algorithms for identifying and clustering in high dimensions [2, 6, 16, 11, 1, 4]. The problem of *clustering* is that of partitioning the points into two sets, with the hope that the points in each set are drawn from different Gaussians. This task generally requires the Gaussians to have little *overlap* (statistical distance near 1); in many such cases they were able to find computationally efficient algorithms for GMMs of more than two Gaussians.

Recall that our goal is to learn the mixture $F = \sum_i w_i F_i$ on a *component by component* basis. An easier problem which has also received some attention is that of learning the *density function* of the entire mixture F without trying to figure out the parameters of the individual components. Recently, a polynomial-time density estimation algorithm was given for *axis-aligned* GMMs, without any nonoverlap assumption [8].

Finally, we note that there is a vast literature on heuristics for the problem of learning GMMs, such as the *EM algorithm* [7]. Our focus in this paper is on algorithms with *provable*

Euclidean distance, in the case where all parameters are bounded.

Runtime is measured in the Real RAM model of computing where arithmetic operations are performed on real numbers.

Density estimation refers to the easier problem of approximating the overall density without necessarily well-approximating individual Gaussians, and axis-aligned Gaussians are those whose principal axes are parallel to the coordinate axes.

guarantees. Even though heuristics such as the EM algorithm often work well in practice, these approaches are not guaranteed to converge to the true parameters. Even worse, the EM algorithm (even for univariate mixtures of just two Gaussians) has been observed to converge very slowly (see Redner and Walker for a thorough treatment [13]) if the algorithm is initialized from a bad starting point.

3. A SIMPLE ALGORITHM

We start by describing a simple algorithm that illustrates our approach to learning GMMs. While the performance guarantees of this algorithm are slightly weaker than those of the full algorithm, described in Section 5, the mechanics of the reduction of the high–dimensional problem into a series of one–dimensional learning problems is made clear.

3.1 A simple one–dimensional algorithm

One would expect that the problem of (approximately) determining the parameters of a univariate mixture of Gaussians should be algorithmically easy because we can resort to a brute force search. However, the surprising difficulty is in proving that a brute force search over a coarse grid of the parameters does not return wrong estimates. For example, what if there are two distinct (univariate) mixtures of Gaussians that do not have *close* parameters, but are nevertheless extremely close as distributions in statistical distance? The central issue in proving that a brute force search does work is in proving that this hypothetical example cannot arise—that two mixtures of (univariate) Gaussians that are not close in parameters must be noticeably different.

To prove this we appeal to the *method of moments*, first introduced by Pearson. We prove that mixtures of k Gaussians are "polynomially robustly identifiable" — that is if two mixtures (of k and $k' \leq k$) Gaussians have sufficiently different parameters, then one of the low-order moments will be noticeably different. In fact, one of the first $4k-2$ moments will be noticeably different, and there are distinct mixtures of k Gaussians that can match exactly on the first linearly (in k) many moments.

Our one–dimensional learning algorithm will draw polynomially many samples from a GMM with k components, and compute the first $4k-2$ moments of this set of samples. We refer to these as the *empirical moments*. Then for every candidate set of parameters $(\{\mu_1, \sigma_1^2, w_1\}, \ldots, (\{\mu_k, \sigma_k^2, w_k\})$ in our grid, we analytically compute the first $4k-2$ moments of the corresponding distribution and compare these values to the empirical moments. If each of the first $4k-2$ analytically computed moments is close enough to the corresponding empirical moment, then we declare success and output that candidate set of parameters.

Through elementary concentration bounds we can demonstrate that, whatever the true parameters are, the closest set of parameters in the grid will pass the test. Thus the algorithm will always output a set of parameter. The correctness of the algorithm then rests on demonstrating that the algorithm will never output a set of parameters that is too far from the true set of parameters. Section 4—showing the polynomially robust identifiability of GMMs—is dedicated to establishing the correctness of this algorithm. Notice that in the case of mixtures of two Gaussians, this brute-force search considers only the first *six* moments and hence we are able to obtain *provable* guarantees for (a variant of) Pearson's *Sixth Moment Test*.

THE SIMPLE 1-D ALGORITHM

Given a set of samples $x_1, \ldots, x_n$ from a one-dimensional GMM with at most k components, and target accuracy ϵ:

- Compute the first $4k-2$ sample moments.
- Calculate the first $4k-2$ moments for every set of parameters, $(\{\mu_1, \sigma_1, w_1\}, \ldots, \{\mu_k, \sigma_k, w_k\})$ such that
  - $w_i \in [0, 1]$,
  - $\mu_i \in [\min_j x_j, \max_j x_j]$,
  - $\sigma_k \in [0, \max_j x_j]$,

  where all parameters are multiples of ϵ^c, where c is a constant dependent on k.
- Return the parameter set whose first $4k-2$ moments most closely match those of the samples.

3.2 A simple high–dimensional algorithm

Our strategy is to obtain, for each component in the mixture, an estimate of this component's mean and variance when projected on many different directions. We can then use these estimates to set up a system of linear constraints on the high–dimensional mean and co-variance matrix of this component, and if we back-solve this system we will obtain good estimates for these high–dimensional parameters.

We choose a vector v uniformly at random from the unit sphere and d^2 perturbations $v_{1,1} \ldots, v_{d,d}$ of v. For each direction $v_{i,j}$, we project the mixture onto direction $v_{i,j}$ and run our one–dimensional learning algorithm. Hence we obtain a set of d^2 one–dimensional mixtures of k Gaussians. But now we are faced with a labeling problem. These d^2 one–dimensional mixtures contain all the information we need: for each high–dimensional Gaussian, we have information about the projection of this Gaussian onto d^2 different directions. In fact, for each high–dimensional Gaussian this information is captured by a representative component from each of the d^2 one–dimensional mixtures. But how do we select out one representative component from each of these d^2 one–dimensional mixtures that together correspond to the projections of a single high–dimensional Gaussian onto d^2 different directions?

In our full-algorithm, we will not always be able to do this. Yet in our simple high–dimensional learning algorithm, we will assume a certain technical condition that will make consistent labeling easy. Specifically, we assume that all components in the original mixture have noticeably different parameters. We can prove that if each pair of components – say, $\mathcal{N}(\mu, \Sigma)$ and $\mathcal{N}(\mu', \Sigma')$ – has either $||\mu - \mu'|| > \epsilon$, or $||\Sigma - \Sigma'||_F > \epsilon$, then with high probability over a randomly chosen direction v, the projected means or projected variances will differ by at least $\text{poly}(\epsilon, \frac{1}{d})$. Intuitively, this makes consistent labeling easy because when we project our data onto v, each component is noticeably different. Yet each $v_{i,j}$ is close to v and hence the projection of any component does not change much from v to $v_{i,j}$.

After re-labeling, for each component in the original mixture we have an estimate of this component's mean and variance when projected onto each direction $v_{i,j}$. The one–dimensional parameters of the projection of a Gaussian are related to the high–dimensional parameters by a system of

linear equations. So for each component in the mixture, we can set up a system of linear equations that constrain the high–dimensional parameters of this component. If our one–dimensional estimates were not *estimates*, but were *exact* then we could back-solve this system to obtain the *exact* high–dimensional parameters. We bound the condition number of this system of equations and so even if our one–dimensional estimates are not exact, but are still good estimates, we will still obtain good estimates for the high–dimensional parameters.

THE SIMPLE HIGH DIMENSIONAL ALGORITHM

Given a set of sample from a GMM in d dimensions with at most k components, target accuracy and probability of failure ϵ, δ:

Let $\epsilon_2 = \left(\frac{\epsilon\delta}{d}\right)^{10}$, $\epsilon_3 = (\epsilon_2)^{10}$.

- Choose vector $v_{0,0}$ uniformly at random from the unit sphere.
- For $i \in \{1, \ldots, d\}$, let e_i denote the basis vector whose ith coordinate is 1, and all other coordinates are 0. For all pairs $i, j \in \{1, \ldots, d\}$, let $v_{i,j} := v + \epsilon_2(e_i + e_j)$.
- For all pairs i, j, project the samples onto $v_{i,j}$, run the *Simple one-dimensional Algorithm* on the resulting one-dimensional data with target accuracy ϵ_3, and let $P_{i,j} := (\{\mu_1, \sigma_1, w_1\}, \ldots)$ be the returned parameters.
- For each $m = 1, 2, \ldots, k$ let $\mu_m^{(0,0)}, \sigma_m^{(0,0)}, w_m^{(0,0)}$ be the recovered parameters of the mth component of $P_{0,0}$. For each pair $i, j \geq 1$ let $\mu_m^{(i,j)}, \sigma_m^{(i,j)}, w_m^{(i,j)}$ be the recovered parameters from $P_{i,j}$ of the component whose parameters are closest, in Euclidean distance, to $(\mu_m^{(0,0)}, \sigma_m^{(0,0)}, w_m^{(0,0)})$.
- For each $m = 1, \ldots, k$, let $w_m = mean(w_m^{(i,j)})$, let μ_m be the point in $\mathbb{R}^d$ whose projections onto $v_{i,j}$ minimize the sum of the squared discrepancies with $\mu_m^{(i,j)}$, and let Σ_m be the positive $d \times d$ semidefinite matrix that minimizes the sum of the squared discrepancies between $v_{i,j}^\dagger \Sigma_m v_{i,j}$ and $(\sigma_m^{(i,j)})^2$.
- If $w_i < \epsilon$, disregard component i.

3.3 Performance of the simple algorithm

The following proposition characterizes the performance of the *Simple High Dimensional Algorithm* described above.

PROPOSITION 1. *Given n independent samples from a GMM $F = \sum_{i=1}^{k'} w_i F_i$ of $k' \leq k$ components in d dimensions with condition number $\kappa(F) > \epsilon$, with probability at least $1 - \delta$, the simple high dimensional algorithm, when run on inputs k, ϵ, δ and the n samples, will return a GMM $\hat{F} = \sum_{i=1}^{k'} w_i F_i$ such that there exists a labeling of the components such that for all i, $|w_i - \hat{w}_i| \leq \epsilon$, $||\mu_i - \hat{\mu}_i|| \leq \epsilon$, and $||\Sigma_i - \hat{\Sigma}_i||_F \leq \epsilon$, provided:*

- $n > \left(\frac{d}{\epsilon\delta\kappa(F)}\right)^{c_k}$,
- *for all $i, j \leq k'$,*

$$|w_i - w_j| + ||\mu_i - \mu_j|| + ||\Sigma_i - \Sigma_j||_F \geq \epsilon.$$

- $|\mu_i|, ||\Sigma_i|| \leq 1/\epsilon$.

The runtime of the estimation algorithm is polynomial in its input size.

The key distinction between the performance of this algorithm, and the performance of the more general algorithm that establishes Theorem 1 is in terms of the distance metric. Here, the input mixture is required to have components whose *parameters* differ from each other by at least ϵ. Our more general algorithm performs on samples from GMMs whose components can have arbitrarily similar parameters, provided that the statistical distance between components is at least ϵ. Additionally, the mixture returned by this simple algorithm is only guaranteed to have have parameters that are very close in Euclidean distance to the true parameters—the mixture returned by the more general algorithm is guaranteed to be very close in both parameter distance and statistical distance.

4. POLYNOMIALLY ROBUST IDENTIFIABILITY

It is well known that two GMMs whose components differ cannot have identical probability densities (see [14], for example)— this is commonly referred to as "identifiability". Thus, given access to arbitrarily many samples, one *can* recover the parameters of a GMM to any desired accuracy. For the purposes of learning with limited data, however, we require a significantly more robust form of identifiability.

Consider two GMMs F, F' whose parameter sets, up to relabeling, differ by ϵ—that is, no matter how one matches the components of the first GMM with the components of the second, there is some component whose mean, variance, or mixing weight differs by at least ϵ from the corresponding component in the other mixture. Given this discrepancy in parameter sets, what is the statistical distance $D(F, F')$? If the answer is inverse exponential in $1/\epsilon$, then our goal of polynomially efficient learning is hopeless—all algorithms for learning GMMs would necessarily require an amount of data exponential in the desired accuracy. Stated differently, the convergence rate of an optimal estimator for the parameters of a GMM would be at most inverse-logarithmic in the number of samples.

Fortunately, such a dependence is not necessary—we show that GMMs are "polynomially robustly identifiable": if mixtures F, F' have parameter sets that differ by ϵ, and mixing weights at least ϵ, then their statistical distance is bounded below by $poly(\epsilon)$, for any fixed number of components. In fact, we prove this by showing that there must be a $poly(\epsilon)$ discrepancy in the first few moments of F and F'. For mixtures of at most k components, considering the first $4k - 2$ moments is sufficient.

THEOREM 2. *Given one dimensional GMMs, F, F', consisting of at most k components, if $\kappa(F) > \epsilon$, then provided that for all $i \leq 4k - 2$,*

$$|E_{x \leftarrow F}[x^i] - E_{x \leftarrow F'}[x^i]| \leq \epsilon^{c_k},$$

then F, F' have the same number of components, $\overline{k}$, and furthermore, there exists a correspondence between the components of F and F' such that the discrepancy in mean, variance, and mixing weight between corresponding components is at most ϵ.

The above theorem guarantees that if two GMMs have very similar low-order moments, the parameter sets must also be very similar, thereby guaranteeing the correctness of the simple brute-force search algorithm for learning one–dimensional GMMs presented in Section 3.

4.1 Deconvolution and moments

Our proof of the polynomially robust identifiability of GMMs relies on considering what happens if one "deconvolves" the probability density functions of a GMM by a Gaussian of carefully chosen variance. The convolution of two Gaussians is a Gaussian, just as the sum of two normal random variables is normal. Hence, we can also consider the "deconvolution" of the mixture by a Gaussian of variance, say, α—this is a simple operation which subtracts α from the variance of each Gaussian in the mixture: Given a GMM with parameter set $(\{\mu_1, \sigma_1^2, w_1\}, \ldots, \{\mu_k, \sigma_k^2, w_k\})$, we define the α-deconvolved mixture to be the GMM with parameter set $(\{\mu_1, \sigma_1^2 - \alpha, w_1\}, \ldots, \{\mu_k, \sigma_k^2 - \alpha, w_k\})$, provided that $\alpha < \min \sigma_i^2$.

The intuition behind considering this transformed mixture is that by decreasing the variance of each Gaussian component, we are effectively disentangling the mixture by making the components overlap less. To illustrate this intuition, suppose we deconvolve by α that is close to the minimum variance of any component. Unless the smallest variance Gaussian is closely matched (in both mean, variance and mixing weight) by a Gaussian in the other mixture, then the two mixtures will have large statistical distance after deconvolution. If, on the other hand, if the smallest variance Gaussian is closely matched, then this pair of components can be stripped away from the respective GMMs as this pair contributes negligibly to the discrepancy between the two mixtures. We can then proceed by induction.

Unfortunately, the deconvolution transformation does not preserve the statistical distance between two distributions. However, we show that it, at least roughly, preserves the disparity in low-order moments of the distributions. Specifically, letting $\mathcal{F}_\alpha(F)$ denote the result of α–deconvolving mixture F, we show that if there is an $i \leq 4k-2$ such that the i^{th} moment of $\mathcal{F}_\alpha(F)$ is at least poly(ϵ) different than the i^{th} moment of $\mathcal{F}_\alpha(F')$ then there is a $j \leq 4k-2$ such that the j^{th} moment of F is at least poly(ϵ) different than the j^{th} moment of F'. To simplify notation, let $M_i[F] = E_{x \leftarrow F}[x^i]$.

LEMMA 3. *Suppose that each constituent Gaussian in F or F' has variances in the interval $[\alpha, 1]$. Then*

$$\frac{\sum_{i=1}^{r} |M_i(\mathcal{F}_\alpha(F)) - M_i(\mathcal{F}_\alpha(F'))|}{\sum_{i=1}^{r} |M_i(F) - M_i(F')|} \leq \frac{(r+1)!}{\lfloor r/2 \rfloor !},$$

The key observation here is that the moments of F and $\mathcal{F}_\alpha(F)$ are related by a simple linear transformation, which can also be viewed as a recurrence relation for Hermite polynomials

4.2 Discrepancy in moments

The deconvolution operation gives us a method to produce a large statistical distance between two mixtures with sufficiently different parameters. Additionally, deconvolution approximately preserves discrepancy in low-order moments. So all that remains is to demonstrate that two mixtures (after an appropriate deconvolution) not only have large statistical distance, but also have a non-negligible discrepancy in moments.

To accomplish this, we show that there are at most $4k-2$ zero-crossings of the difference in densities, $f = \mathcal{F}_\alpha(F) - \mathcal{F}_\alpha(F')$ and then we construct a degree $4k-2$ polynomial $p(x)$ that always has the same sign as $f(x)$, so that when $p(x)$ is integrated against $f(x)$ the result is at least poly(ϵ). We construct this polynomial so that the coefficients are bounded, and this implies that there is some moment i (at most the degree of the polynomial) for which the difference between the i^{th} raw moment of $\mathcal{F}_\alpha(F)$ and of $\mathcal{F}_\alpha(F')$ is large.

The first step is to show that the point-wise difference between the density functions of any two mixtures of k Gaussians is either identically zero, or has at most $4k-2$ zero crossings. This bound can be easily shown to be tight. Our proof of this fact relies on the following Theorem, due to Hummel and Gidas [10]:

THEOREM 4 (THM 2.1 IN [10]). *Given $f(x) : \mathbb{R} \to \mathbb{R}$, that is analytic and has n zeros, then for any $\sigma^2 > 0$, the function $g(x) = f(x) \circ \mathcal{N}(0, \sigma^2, x)$ has at most n zeros.*

Given this theorem, we can then prove an upper bound on the number of zero crossings by isolating the smallest variance component through deconvolution, removing this component and then proceeding inductively; by the above theorem, deconvolution does not decrease the number of zero crossings and since each component that we remove in this way is essentially a delta function, its removal reduces the number of zero crossings by at most two.

The proof of Theorem 2 then follows from assembling the pieces: Given a pair of one–dimensional GMMs whose parameter sets differ (1) strip away all components in pairs of closely corresponding components, which has negligible effect on the discrepancy in moments of the the pair of GMMs; (2) deconvolve by nearly the variance of the skinniest remaining component; (3) since this component is now nearly a Dirac delta function and there is no closely matching component in the second GMM, the deconvolved GMMs have non-negligible statistical distance; (4) non-negligible statistical distance implies non-negligible moment discrepancy; and (5) if there is a discrepancy in one the low-order moments of two GMMs, then after convolution by a Gaussian, there will still be a discrepancy in some low-order moment.

5. THE FULL ALGORITHM

We now motivate and describe our general algorithm for learning GMMs which, with high probability, returns a mixture whose components are accurate in terms of statistical distance. To get an intuitive sense for the types of mixtures for which the simple high–dimensional algorithm fails, consider the mixture of three components depicted in Figure 3. The two narrow components are very similar: both their means, and their covariance matrices are nearly identical. With overwhelming probability, the projection of this mixture onto any one–dimensional space will result in these two components becoming indistinguishable given any reasonable amount of data. Nevertheless, the statistical distance between these two components is close to one, and thus, information theoretically, we *should* be able to distinguish them.

How can we hope to disentangle these two components if, in nearly every one–dimensional projection, these components are indistinguishable? The intuition for the solution

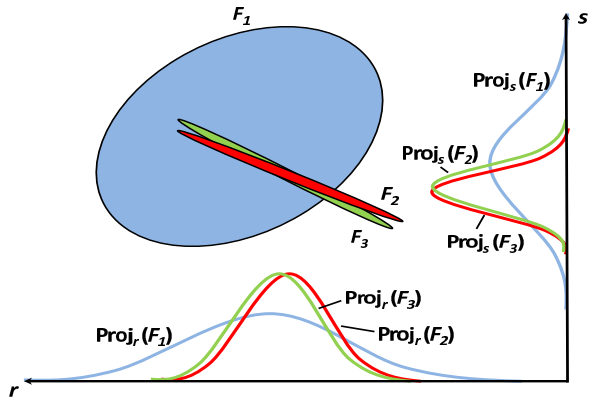

Figure 3: An example of a GMM with three components F_1, F_2, F_3, such that with high probability over random vectors, the one dimensional projections of F_2 and F_3 will be very similar, despite $D_{tv}(F_2, F_3) \approx 1$.

is also provided in the example: we can cluster out these two components and recurse. In particular, there is a vector (corresponding to the direction of small variance of these two components) such that if we project all the data onto this direction, the pair of narrow Gaussians are almost completely "disentangled" from the third component. Almost all of the data corresponding to the two narrow components will be contained within a small interval when projected on this direction, and almost none of the data generated by the third component will be contained in this interval.

If we are able to successfully perform such a clustering of the original mixture into two sub-mixtures, we can recurse. The central insight is that if we consider the sub-mixture corresponding to just the two narrow Gaussians, then we can re-scale the space by applying an affine transformation so that the resulting mean and variance are zero and one, respectively, in every direction. This re-scaling has the effect of stretching out this sub-mixture along the direction of small variance. In this resulting mixture of two Gaussians, if we project on a randomly chosen direction, the components *will* be noticeably different.

Our full algorithm will follow this general plan—in each step, our algorithm either learns a good estimate and outputs this estimate, or else will cluster the mixture into two proper sub-mixtures and recurse. The remainder of this section is devoted to explaining how we can learn a direction of small variance, and hence enable the clustering and recursion step if we are not able to directly apply the *Simple High Dimensional Algorithm* of Section 3 to learn good estimates for the GMM components.

5.1 Learning a direction to project

The main remaining question is how to find a vector in which some of the components have small variance. Intuitively, finding this direction seems to require knowledge of the true mixture. Our approach will be to learn an estimate that is close to some *partition* of the true components.

Suppose we add d-dimensional Gaussian noise to samples drawn from the example GMM of Figure 3. This would have the effect of "fattening" each component. After "fattening", the two narrow components would have extremely small statistical distance. So we could run our simple learning algorithm on this "fattened" mixture. Even though this distribution is a mixture of three Gaussians, the mixture is statistically extremely close to a mixture of two Gaussians. Our simple learning algorithm will return an estimate mixture of two Gaussians with the property that each component is close to a sub-mixture of the "fattened" distribution.

Thus one of the components in this estimate will correspond to the sub-mixture of the two narrow components. By examining this component, we notice that it is "skinny" (after adjusting the covariance matrix to account for the noise that we artificially added). Hence if we compute the smallest eigenvector of this co-variance matrix, we recover a direction which allows us to cluster the original mixture into two sub-mixtures and recurse.

THE FULL HIGH DIMENSIONAL ALGORITHM

Given a set of sample from a GMM in d dimensions with at most k components, and target accuracy ϵ: Let $\epsilon_2 = \epsilon^c$, (for a constant c dependent on k).

- Rescale the set of samples so as to have mean 0 and covariance the identity matrix.
- Create a *fattened* set of samples: for each of the original samples add an independent $x \leftarrow \mathcal{N}(0, I_{d\times d})$.
- Define $\epsilon_1, \ldots, \epsilon_{k^2}$ with $\epsilon_i = \epsilon^{c' \cdot i}$, for a constant c' (dependent on k.) Run the *Simple High Dimensional Algorithm* on the fattened samples with each target accuracy ϵ_i, yielding k^2 parameter sets $P_1, \ldots, P_k$.
- Find a *consistent* chain of at least $k^2/2$ parameter sets; we say P_i is consistent with P_j for $i < j$ if there exists a mapping of the components of P_i into the components of P_j such that the statistical distance between each component of P_i and its image in P_j is at most $\epsilon_i + \epsilon_j$.
- Let $P' = (\{\mu_1, \Sigma_1, w_1\}, \ldots)$ be one of these parameter sets in the chain, and let $P = (\{\mu_1, \Sigma_1 - I, w_1\}, \{\mu_2, \Sigma_2 - I, w_2\}, \ldots)$ be the *unfattened* parameters.
- Let $k' \leq k$ be the number of components of P. Let λ be the minimum over $i \in \{1, \ldots, k'\}$, of the minimum eigenvalue of Σ_i.
  - If $\lambda > \epsilon_2$, output the recovered parameters and return $SUCESS$.
  - Otherwise, project the original (non-noisy) samples onto the eigenvector corresponding to this minimum eigenvalue, and cluster the samples into two clusters, with one cluster corresponding to samples that likely originated from the component with the smallest minimum eigenvalue.
  - Recursively apply this entire algorithm to each of the two sets of samples, with target accuracy ϵ, and number of components at most $k - 1$.

If the *Simple High Dimensional Algorithm* is run on samples from a GMM in which all components have large minimum eigenvalue (for example, if the samples have been "fattened"), then the algorithm, when run with target accuracy

ϵ, will successfully learn the mixture provided that for each pair of components, either the statistical distance is at least ϵ, or at most $\epsilon' << \epsilon$, where $\epsilon' = p(\epsilon)$ for some polynomial p. In the case that some set of components all have pairwise statistical distance at most ϵ', then the simple high dimensional algorithm will never realize that these components correspond to separate components, and will simply return a single component in place of this set. The difficulty is when there exists some pair of components whose statistical distance lies within this bad *window* $[p(\epsilon), \epsilon]$. In such an instance, the *Simple High Dimensional Algorithm* has no provable guarantees.

To avoid the potential difficulty of finding a target accuracy ϵ for which no pair of components have statistical distance within the associated inadmissable window, one simply runs the high dimensional algorithm with a range of target accuracies, $\epsilon_1, \ldots, \epsilon_{k^2}$, with $\epsilon_i < p(\epsilon_{i-1})$. While we will never know which runs succeeded, there are at most $\binom{k}{2}$ pairwise statistical distances, and each pairwise statistical distance can fall into the inadmissible window of at most one run, and thus a majority of the runs will be successful. All that remains is to find a set of at least $k^2/2$ runs which are consistent: given two sets of parameters returned by runs with target accuracies $\epsilon_1 < \epsilon_2$, we say they are consistent if there is some surjective mapping of the components returned by the ϵ_1 run into the components returned by the ϵ_2 run, such that each component has similar mean and covariance to its image. Thus, one can find such a chain of at least $k^2/2$ consistent runs, yielding a set of accurate parameters.

6. EXPONENTIAL DEPENDENCE ON K

While the dependency of our algorithm on the number of components is super–exponential, we also give a lower bound that shows that at least an exponential dependency is necessary, even for mixtures in just one dimension. We show this by giving an explicit construction, for any k, of two one–dimensional GMMs F_1, F_2 consisting of at most k Gaussian components where the mixing weights and pairwise statistical distances between components are at least $1/2k$. Additionally, in any correspondence between the components of one mixture, and the components of the other, there is at least one component in F_1 whose mean or variance differs from the corresponding component in F_2 by at least 1. Nevertheless, $D_{tv}(F_1, F_2) < 1/e^{O(k)}$, and thus it is information theoretically impossible to distinguish a set of $e^{o(k)}$ samples from F_1 from a set of $e^{o(k)}$ samples from F_2, and hence impossible to return the component parameters to within $\pm 1/2$ (with high probability).

THEOREM 5. *There exists a pair of GMMs, F_1, F_2 with at most k components each and condition numbers at least $1/2k$ such $D_{tv}(F_1, F_2) = 1/e^{O(k)}$, yet for any mapping between the components of F_1 and F_2, there will be a component whose variance differs by at least 1 from that of its image.*

The construction hinges on the inverse exponential in k statistical distance between $\mathcal{N}(0, 2)$, and a mixture of infinitely many Gaussians of unit variance whose components are centered at multiples of $1/\sqrt{k}$, where the weight assigned to the component centered at $i/\sqrt{k}$ is given by $N(0, 1, i/\sqrt{k})$. Verifying this claim is an exercise in Fourier analysis. We then modify the construction slightly so that both GMMs have at most k components, and so that all components having weight at least $2/k$.

7. CONCLUSIONS

The primary contribution of our research is to get a first handle on the *sample complexity* and *computational complexity* of this problem—how many samples are required to learn, how much runtime is necessary, and in which parameters are these exponential or polynomial? While we do not know the optimal achievable rates, distinguishing between polynomial and exponential is a telling start.

Asymptotic guarantees are merely a guide as to which algorithms may perform well. Our current algorithm is not designed to be practical in any meaningful sense. However, we hope it opens the door to future work on algorithms that are both of practical utility and theoretically motivated, i.e., efficient estimators which do not suffer from local minima.

8. REFERENCES

[1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *COLT*, pages 458–469, 2005.

[2] S. Arora and R. Kannan. Learning mixtures of arbitrary Gaussians. In *STOC*, pages 247–257, 2001.

[3] J. Brainard and D. E. Burmaster. Bivariate distributions for height and weight of men and women in the united states. *Risk analysis an official publication of the Society for Risk Analysis*, 12(2):267–275, 1992.

[4] S. C. Brubaker and S. Vempala. Isotropic pca and affine-invariant clustering. In *FOCS*, pages 551–560, 2008.

[5] S. Dasgupta. Learning mixtures of Gaussians. In *FOCS*, pages 634–644, 1999.

[6] S. Dasgupta and L. J. Schulman. A two-round variant of EM for Gaussian mixtures. In *UAI*, pages 152–159, 2000.

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

[8] J. Feldman, R. A. Servedio, and R. O'Donnell. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *COLT*, pages 20–34, 2006.

[9] D. Hsu, S. M. Kakade, and T. Zhang. A spectral algorithm for learning hidden Markov models. In *COLT*, 2009.

[10] R. A. Hummel and B. C. Gidas. Zero crossings and the heat equation. *Technical Report Number 111, Courant Institute of Mathematical Sciences at NYU*, 1984.

[11] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. *SIAM J. Comput.*, 38(3):1141–1156, 2008.

[12] K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London*, pages 71–110, 1894.

[13] R. A. Redner and H. F. Walker. Mixture densities, maximum likelihood and the EM algorithm. *SIAM Review*, 26:195–239, 1984.

[14] H. Teicher. Identifiability of mixtures. *Ann. Math. Statist.*, 32(1):244–248, 1961.

[15] S. Vempala. The random projection method. *American Mathematical Society*, 2004.

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