Learning Mixtures of Gaussians in High Dimensions

[Extended Abstract]*

Rong Ge
Microsoft Research,
New England

Qingqing Huang
MIT, EECS

Sham M. Kakade
Microsoft Research,
New England

ABSTRACT

Efficiently learning mixture of Gaussians is a fundamental problem in statistics and learning theory. Given samples coming from a random one out of k Gaussian distributions in $\mathbb{R}^n$, the learning problem asks to estimate the means and the covariance matrices of these Gaussians. This learning problem arises in many areas ranging from the natural sciences to the social sciences, and has also found many machine learning applications.

Unfortunately, learning mixture of Gaussians is an information theoretically hard problem: in order to learn the parameters up to a reasonable accuracy, the number of samples required is exponential in the number of Gaussian components in the worst case. In this work, we show that provided we are in high enough dimensions, the class of Gaussian mixtures is learnable in its most general form under a smoothed analysis framework, where the parameters are randomly perturbed from an adversarial starting point.

In particular, given samples from a mixture of Gaussians with randomly perturbed parameters, when $n \geq \Omega(k^2)$, we give an algorithm that learns the parameters with polynomial running time and using polynomial number of samples.

The central algorithmic ideas consist of new ways to decompose the moment tensor of the Gaussian mixture by exploiting its structural properties. The symmetries of this tensor are derived from the combinatorial structure of higher order moments of Gaussian distributions (sometimes referred to as Isserlis' theorem or Wick's theorem). We also develop new tools for bounding smallest singular values of structured random matrices, which could be useful in other smoothed analysis settings.

Keywords

mixture models; spectral methods; smoothed analysis.

*A full version of this paper is available at `http://arxiv.org/abs/1503.00424`. The Second author was supported in part by NSF/CPS 6924594

STOC'15, June 14–17, 2015, Portland, Oregon, USA.

http://dx.doi.org/10.1145/2746539.2746616 .

1. INTRODUCTION

Learning mixtures of Gaussians is a fundamental problem in statistics and learning theory, whose study dates back to Pearson (1894). Gaussian mixture models arise in numerous areas including physics, biology and the social sciences (McLachlan and Peel (2004); Titterington et al. (1985)), as well as in image processing (Reynolds and Rose (1995)) and speech (Permuter et al. (2003)).

In a Gaussian mixture model, there are k unknown n-dimensional multivariate Gaussian distributions. Samples are generated by first picking one of the k Gaussians, then drawing a sample from that Gaussian distribution. Given samples from the mixture distribution, our goal is to estimate the means and covariance matrices of these underlying Gaussian distributions[1].

This problem has a long history in theoretical computer science. The seminal work of Dasgupta (1999) gave an algorithm for learning spherical Gaussian mixtures when the means are well separated. Subsequent works (Dasgupta and Schulman (2000); Sanjeev and Kannan (2001); Vempala and Wang (2004); Brubaker and Vempala (2008)) developed better algorithms in the well-separated case, relaxing the spherical assumption and the amount of separation required.

When the means of the Gaussians are not separated, after several works (Belkin and Sinha (2009); Kalai et al. (2010)), Belkin and Sinha (2010) and Moitra and Valiant (2010) independently gave algorithms that run in polynomial time and with polynomial number of samples for a fixed number of Gaussians. However, both running time and sample complexity depend *super* exponentially on the number of components k^2. Their algorithm is based on the *method of moments* introduced by Pearson (1894): first estimate the $O(k)$-order moments of the distribution, then try to find the parameters that agree with these moments. Moitra and Valiant (2010) also show that the exponential dependency of the sample complexity on the number of components is necessary, by constructing an example of two mixtures of Gaussians with very different parameters, yet with exponentially small statistical distance.

Recently, Hsu and Kakade (2013) applied spectral methods to learning mixture of spherical Gaussians. When $n \geq k + 1$ and the means of the Gaussians are linearly independent, their algorithm can learn the model in polynomial time and with polynomial number of samples. This

[1] This is different from the problem of *density estimation* considered in Feldman et al. (2006); Chan et al. (2014)

[2] In fact, it is in the order of $O(e^{O(k)^k})$ as shown in Theorem 11.3 in Valiant (2012).

result suggests that the lower bound example in Moitra and Valiant (2010) is only a *degenerate* case in high dimensional space. In fact, *most* (in general position) mixture of spherical Gaussians are *easy* to learn. This result is also based on the method of moments, and only uses second and third moments. Several follow-up works (Bhaskara et al. (2014); Anderson et al. (2013)) use higher order moments to get better dependencies on n and k.

However, the algorithm in Hsu and Kakade (2013) as well as in the follow-ups all make strong requirements on the covariance matrices. In particular, most of them only apply to learning mixture of spherical Gaussians. For mixture of Gaussians with general covariance matrices, the best known result is still Belkin and Sinha (2010) and Moitra and Valiant (2010), which algorithms are not polynomial in the number of components k. This leads to the following natural question:

Question: *Is it possible to learn* most *mixture of Gaussians in polynomial time using a polynomial number of samples?*

Our Results.

In this paper, we give an algorithm that learns *most* mixture of Gaussians in high dimensional space (when $n \geq \Omega(k^2)$), and the argument is formalized under the *smoothed analysis* framework first proposed in Spielman and Teng (2004).

In the smoothed analysis framework, the adversary first choose an arbitrary mixture of Gaussians. Then the mean vectors and covariance matrices of this Gaussian mixture are randomly *perturbed* by a small amount ρ [3]. The samples are then generated from the Gaussian mixture model with the perturbed parameters. The goal of the algorithm is to learn the perturbed parameters from the samples.

The smoothed analysis framework is a natural bridge between worst-case and average-case analysis. On one hand, it is similar to worst-case analysis, as the adversary chooses the initial instance, and the perturbation allowed is small. On the other hand, even with small perturbation, we may hope that the instance be different enough from degenerate cases. A successful algorithm in the smoothed analysis setting suggests that the bad instances must be very "sparse" in the parameter space: they are highly unlikely in any small neighborhood of any instance. Recently, the smoothed analysis framework has also motivated several research work (Kalai et al. (2009) Bhaskara et al. (2014)) in analyzing learning algorithms.

In the smoothed analysis setting, we show that it is easy to learn most Gaussian mixtures:

THEOREM 1.1. *(informal statement of Theorem 3.4) In the smoothed analysis setting, when $n \geq \Omega(k^2)$, given samples from the perturbed n-dimensional Gaussian mixture model with k components, there is an algorithm that learns the correct parameters up to accuracy ϵ with high probability, using polynomial time and number of samples.*

An important step in our algorithm is to learn Gaussian mixture models whose components all have mean zero, which is also a problem of independent interest (Zoran and Weiss (2012)). Intuitively this is also a "hard" case, as there is no separation in the means. Yet algebraically, this case gives rise to a novel tensor decomposition algorithm. The ideas for solving this decomposition problem are then generalized to tackle the most general case.

THEOREM 1.2. *(informal statement of Theorem 3.5) In the smoothed analysis setting, when $n \geq \Omega(k^2)$, given samples from the perturbed mixture of* zero-mean *n-dimensional Gaussian mixture model with k components, there is an algorithm that learns the parameters up to accuracy ϵ with high probability, using polynomial running time and number of samples.*

Organization.

The main part of the paper will focus on learning mixtures of zero-mean Gaussians. The proposed algorithm for this special case contains most of the new ideas and techniques. In Section 2 we introduce the notations for matrices and tensors which are used to handle higher order moments throughout the discussion. Then in Section 3 we introduce the smoothed analysis model for learning mixture of Gaussians and discuss the moment structure of mixture of Gaussians, then we formally state our main theorems. Section 4 outlines our algorithm for learning zero-mean mixture of Gaussians. In Section 6 we briefly discuss how the ideas for zero-mean case can be generalized to learning mixture of nonzero Gaussians.

2. NOTATIONS

Vectors and Matrices.

In the vector space $\mathbb{R}^n$, let $\langle \cdot, \cdot \rangle$ denote the inner product of two vectors, and $\| \cdot \|$ to denote the Euclidean norm.

For a tall matrix $A \in \mathbb{R}^{m \times n}$, let $A_{[:,j]}$ denote its j-th column vector, let $A^\top$ denote its transpose, $A^\dagger = (A^\top A)^{-1} A^\top$ denote the pseudoinverse, and let $\sigma_k(A)$ denote its k-th singular value. Let I_n be the identity matrix of dimension $n \times n$. The spectral norm of a matrix is denoted as $\| \cdot \|$, and the Frobenius norm is denoted as $\| \cdot \|_F$. We use $A \succeq 0$ for positive semidefinite matrix A.

In the discussion, we often need to convert between vectors and matrices. Let $\mathrm{vec}(A) \in \mathbb{R}^{mn}$ denote the vector obtained by stacking all the columns of A. For a vector $x \in \mathbb{R}^{m^2}$, let $\mathrm{mat}(x) \in \mathbb{R}^{m \times m}$ denote the inverse mapping such that $\mathrm{vec}(\mathrm{mat}(x)) = x$.

We use $[n]$ to denote the set $\{1, 2, ..., n\}$ and $[n] \times [n]$ to denote the set $\{(i, j) : i, j \in [n]\}$. These are often used as indices of matrices.

Symmetric matrices.

We use $\mathbb{R}^{n \times n}_{sym}$ to denote the space of all $n \times n$ symmetric matrices, which subspace has dimension $\binom{n+1}{2}$. Since we will frequently use $n \times n$ and $k \times k$ symmetric matrices, we denote their dimensions by the constants $n_2 = \binom{n+1}{2}$ and $k_2 = \binom{k+1}{2}$. Similarly, we use $\mathbb{R}^{n \times \cdots \times n}_{sym}$ to denote the symmetric k-dimensional multi-arrays (tensors), which subspace has dimension $\binom{n+k-1}{k}$. If a k-th order tensor $X \in \mathbb{R}^{n \times \cdots \times n}_{sym}$, then for any permutation π over $[k]$, we have $X_{n_1,...,n_k} = X_{n_{\pi(1)},...,n_{\pi(k)}}$.

[3] See Definition 3.2 in Section 3.1 for the details.

Linear subspaces.

We represent a linear subspace $\mathcal{S} \in \mathbb{R}^n$ of dimension d by a matrix $S \in \mathbb{R}^{n\times d}$, whose columns of S form an (arbitrary) orthonormal basis of the subspace. The projection matrix onto the subspace $\mathcal{S}$ is denoted by $\mathrm{Proj}_S = SS^\top$, and the projection onto the orthogonal subspace $\mathcal{S}^\perp$ is denoted by $\mathrm{Proj}_{S^\perp} = I_n - SS^\top$. When we talk about the span of several matrices, we mean the space spanned by their vectorization.

Tensors.

A tensor is a multi-dimensional array. Tensor notations are useful for handling higher order moments. We use $\otimes$ to denote tensor product, suppose $a, b, c \in \mathbb{R}^n$, $T = a \otimes b \otimes c \in \mathbb{R}^{n\times n\times n}$ and $T_{i_1,i_2,i_3} = a_{i_1} b_{i_2} c_{i_3}$. For a vector $x \in \mathbb{R}^n$, let the t-fold tensor product $x^{\otimes t}$ denote the t-th order rank one tensor $(x^{\otimes t})_{i_1,i_2,\ldots,i_t} = \prod_{j=1}^t x_{i_j}$.

Every tensor defines a multilinear mapping. Consider a 3-rd order tensor $X \in \mathbb{R}^{n_A\times n_B\times n_C}$. For given dimension m_A, m_B, m_C, it defines a multi-linear mapping $X(\cdot,\cdot,\cdot) : \mathbb{R}^{n_A\times m_A} \times \mathbb{R}^{n_B\times m_B} \times \mathbb{R}^{n_C\times m_C} \to \mathbb{R}^{m_A\times m_B\times m_C}$ defined as below: $(\forall j_1 \in [m_A], j_2 \in [m_B], j_3 \in [m_C])$

$$[X(V_1,V_2,V_3)]_{j_1,j_2,j_3} = \sum_{i_1\in[n_A],i_2\in[n_B],i_3\in[n_C]} X_{i_1,i_2,i_3}[V_1]_{j_1,i_1}[V_2]_{j_2,i_2}[V_3]_{j_3,i_3}.$$

If X admits a decomposition $X = \sum_{i=1}^k A_{[:,i]} \otimes B_{[:,i]} \otimes C_{[:,i]}$ for $A \in \mathbb{R}^{n_A\times k}, B \in \mathbb{R}^{n_B\times k}, C \in \mathbb{R}^{n_C\times k}$, the multi-linear mapping has the form $X(V_1,V_2,V_3) = \sum_{i=1}^k (V_1^\top A_{[:,i]}) \otimes (V_2^\top B_{[:,i]}) \otimes (V_3^\top C_{[:,i]})$.

In particular, the vector given by $X(\mathbf{e}_i, \mathbf{e}_j, I)$ is the one-dimensional slice of the 3-way array, with the index for the first dimension to be i and the second dimension to be j.

Matrix Products.

We use $\odot$ to denote column wise Katri-Rao product, and $\otimes_{kr}$ to denote Kronecker product. As an example, for matrices $A \in \mathbb{R}^{m_A\times n}$, $B \in \mathbb{R}^{m_B\times n}$, $C \in \mathbb{R}^{m_C\times n}$:

$$[A\otimes B\otimes C]_{j_1,j_2,j_3} = \sum_{i=1}^n A_{j_1,i}B_{j_2,i}C_{j_3,i},$$

$$[A\odot B]_{[:,j]} = A_{[:,j]} \otimes_{kr} B_{[:,j]},$$

$$A \otimes_{kr} B = \begin{bmatrix} A_{1,1}B & \cdots & A_{1,n}B \\ \vdots & \ddots & \vdots \\ A_{m_A,1}B & \cdots & A_{m_A,n}B \end{bmatrix}.$$

3. MAIN RESULTS

In this section, we first formally introduce the smoothed analysis framework for our problem and state our main theorems. Then we will discuss the structure of the moments of Gaussian mixtures, which is crucial for understanding our method of moments based algorithm.

3.1 Smoothed Analysis for Learning Mixture of Gaussians

Let $\mathcal{G}_{n,k}$ denote the class of Gaussian mixtures with k components in $\mathbb{R}^n$. A distribution in this family is specified by the following parameters: the mixing weights ω_i, the mean vectors $\mu^{(i)}$ and the covariance matrices $\Sigma^{(i)}$, for $i \in [k]$.

$$\mathcal{G}_{n,k} := \Big\{\mathcal{G} = \{(\omega_i, \mu^{(i)}, \Sigma^{(i)})\}_{i\in[k]} : \omega_i \in \mathbb{R}_+,\ \sum_{i=1}^k \omega_i = 1,\ \mu^{(i)} \in \mathbb{R}^n,\ \Sigma^{(i)} \in \mathbb{R}^{n\times n}_{sym},\ \Sigma^{(i)} \succeq 0\Big\}.$$

As an interesting special case of the general model, we also consider the mixture of "zero-mean" Gaussians, which has $\mu^{(i)} = 0$ for all components $i \in [k]$.

A sample x from a mixture of Gaussians is generated in two steps:

1. Sample $h \in [k]$ from a multinomial distribution, with probability $\Pr[h = i] = \omega_i$ for $i \in [k]$.
2. Sample $x \in \mathbb{R}^n$ from the h-th Gaussian distribution $\mathcal{N}(\mu^{(h)}, \Sigma^{(h)})$.

The learning problem asks to estimate the parameters of the underlying mixture of Gaussians:

DEFINITION 3.1 (LEARNING MIXTURE OF GAUSSIANS). *Given N samples $x_1, x_2, ..., x_N$ drawn i.i.d. from a mixture of Gaussians $\mathcal{G} = \{(\omega_i, \mu^{(i)}, \Sigma^{(i)})\}_{i\in[k]}$, an algorithm learns the mixture of Gaussians with accuracy ϵ, if it outputs an estimation $\widehat{\mathcal{G}} = \{(\widehat{\omega}_i, \widehat{\mu}^{(i)}, \widehat{\Sigma}^{(i)})\}_{i\in[k]}$ such that there exists a permutation π on $[k]$, and for all $i \in [k]$, we have $|\widehat{\omega}_i - \omega_{\pi(i)}| \le \epsilon$, $\|\widehat{\mu}^{(i)} - \mu^{(\pi(i))}\| \le \epsilon$ and $\|\widehat{\Sigma}^{(i)} - \Sigma^{(\pi(i))}\| \le \epsilon$.*

In the worst case, learning mixture of Gaussians is a information theoretically hard problem (Moitra and Valiant (2010)). There exists worst-case examples where the number of samples required for learning the instance is at least exponential in the number of components k (McLachlan and Peel (2004)). The non-convexity arises from the hidden variable h: without knowing h we cannot determine which Gaussian component each sample comes from.

The smoothed analysis framework provides a way to circumvent the worst case instances, yet still studying this problem in its most general form. The basic idea is that, with high probability over the small random perturbation to any instance, the instance will not be a "worst-case" instance, and actually has reasonably good condition for the algorithm.

Next, we show how the parameters of the mixture of Gaussians are *perturbed* in our setup.

DEFINITION 3.2 (ρ-SMOOTH MIXTURE OF GAUSSIANS). *For $\rho < 1/n$, a ρ-smooth n-dimensional k-component mixture of Gaussians $\widetilde{\mathcal{G}} = \{(\widetilde{\omega}_i, \widetilde{\mu}^{(i)}, \widetilde{\Sigma}^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$ is generated as follows:*

1. *Choose an arbitrary (could be adversarial) instance $\mathcal{G} = \{(\omega_i, \mu^{(i)}, \Sigma^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$. Scale the distribution such that $0 \preceq \Sigma^{(i)} \preceq \frac{1}{2} I_n$ and $\|\mu^{(i)}\| \le \frac{1}{2}$ for all $i \in [k]$.*
2. *Let $\Delta_i \in \mathbb{R}^{n\times n}_{sym}$ be a random symmetric matrix with zeros on the diagonals, and the upper-triangular entries are independent random Gaussian variables $\mathcal{N}(0, \rho^2)$. Let $\delta_i \in \mathbb{R}^n$ be a random Gaussian vector with independent Gaussian variables $\mathcal{N}(0, \rho^2)$.*
3. *Set $\widetilde{\omega}_i = \omega_i$, $\widetilde{\mu}^{(i)} = \mu^{(i)} + \delta_i$, $\widetilde{\Sigma}^{(i)} = \Sigma^{(i)} + \Delta_i$.*

4. *Choose the diagonal entries of $\widetilde{\Sigma}^{(i)}$ arbitrarily, while ensuring the positive semi-definiteness of the covariance matrix $\widetilde{\Sigma}^{(i)}$, and the diagonal entries are upper bounded by 1. The perturbation procedure fails if this step is infeasible*[4].

A ρ-smooth zero-mean mixture of Gaussians is generated using the same procedure, except that we set $\widetilde{\mu}^{(i)} = \mu^{(i)} = 0$, for all $i \in [k]$.

REMARK 3.3. *When the original matrix is of low rank, a simple random perturbation may not lead to a positive semidefinite matrix, which is why our procedure of perturbation is more restricted in order to guarantee that the perturbed matrix is still a valid covariance matrix.*

There could be other ways of locally perturbing the covariance matrix. Our procedure actually gives more power *to the adversary as it can change the diagonals* after *observing the perturbations for other entries. Note that with high probability if we just let the new diagonal to be $5\sqrt{n}\rho$ larger than the original ones, the resulting matrix is still a valid covariance matrix. In other words, the adversary can always keep the perturbation small if it wants to.*

Instead of the worst-case problem in Definition 3.1, our algorithms work on the smoothed instance. Here the model first gets perturbed to $\widetilde{\mathcal{G}} = \{(\widetilde{\omega}_i, \widetilde{\mu}^{(i)}, \widetilde{\Sigma}^{(i)})\}_{i\in[k]}$, the samples are drawn according to the perturbed model, and the algorithm tries to learn the perturbed parameters. We give a polynomial time algorithm in this case:

THEOREM 3.4 (MAIN THEOREM). *Consider a ρ-smooth mixture of Gaussians $\widetilde{\mathcal{G}} = \{(\widetilde{\omega}_i, \widetilde{\mu}^{(i)}, \widetilde{\Sigma}^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$ for which the number of components is at least* [5] *$k \geq C_0$ and the dimension $n \geq C_1 k^2$, for some fixed constants C_0 and C_1. Suppose that the mixing weights $\widetilde{\omega}_i \geq \omega_o$ for all $i \in [k]$. Given N samples drawn i.i.d. from $\widetilde{\mathcal{G}}$, there is an algorithm that learns the parameters of $\widetilde{\mathcal{G}}$ up to accuracy ϵ, with high probability over the randomness in both the perturbation and the samples. Furthermore, the running time and number of samples N required are both upper bounded by $poly(n, k, 1/\omega_o, 1/\epsilon, 1/\rho)$.*

To better illustrate the algorithmic ideas for the general case, we first present an algorithm for learning mixtures of zero-mean Gaussians. Note that this is not just a special case of the general case, as with the smoothed analysis, the zero mean vectors are not perturbed.

THEOREM 3.5 (ZERO-MEAN). *Consider a ρ-smooth mixture of zero-mean Gaussians $\widetilde{\mathcal{G}} = \{(\widetilde{\omega}_i, 0, \widetilde{\Sigma}^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$ for which the number of components is at least $k \geq C_0$ and the dimension $n \geq C_1 k^2$, for some fixed constants C_0 and C_1. Suppose that the mixing weights $\widetilde{\omega}_i \geq \omega_o$ for all $i \in [k]$. Given N samples drawn i.i.d. from $\widetilde{\mathcal{G}}$, there is an algorithm that learns the parameters of $\widetilde{\mathcal{G}}$ up to accuracy ϵ, with high probability over the randomness in both the perturbation and the samples. Furthermore, the running time and number of samples N are both upper bounded by $poly(n, k, 1/\omega_o, 1/\epsilon, 1/\rho)$.*

Throughout the paper we always assume that $n \geq C_1 k^2$ and $\widetilde{\omega}_i \geq \omega_o$.

3.2 Moment Structure of Mixture of Gaussians

Our algorithm is also based on the method of moments, and we only need to estimate the 3-rd, the 4-th and the 6-th order moments. In this part we briefly discuss the structure of 4-th and 6-th moments in the zero-mean case (3-rd moment is always 0 in the zero-mean case). These structures are essential to the proposed algorithm.

The m-th order moments of the *zero-mean* Gaussian mixture model $\mathcal{G} \in \mathcal{G}_{n,k}$ are given by the following m-th order symmetric tensor $M_m \in \mathbb{R}^{n\times\cdots\times n}_{sym}$: $\forall j_1, \ldots, j_m \in [n]$,

$$[M_m]_{j_1,\ldots,j_m} := \mathbb{E}\left[x_{j_1} \ldots x_{j_m}\right] = \sum_{i=1}^{k} \omega_i \mathbb{E}\left[y^{(i)}_{j_1} \ldots y^{(i)}_{j_m}\right],$$

where $y^{(i)}$ corresponds to the n-dimensional zero-mean Gaussian distribution $\mathcal{N}(0, \Sigma^{(i)})$. The moments for each Gaussian component are characterized by Isserlis's theorem as below:

THEOREM 3.6 (ISSERLIS' THEOREM). *Let $(y_1, \ldots, y_{2t})$ be a multivariate zero-mean Gaussian random vector $\mathcal{N}(0, \Sigma)$, then*

$$\mathbb{E}[y_1 \ldots y_{2t}] = \sum \prod \Sigma_{u,v},$$

where the summation is taken over all distinct ways of partitioning $y_1, \ldots, y_{2t}$ into t pairs, which correspond to all the perfect matchings in a complete graph.

Ideally, we would like to obtain the following quantities (recall $n_2 = \binom{n+1}{2}$):

$$X_4 = \sum_{i=1}^{k} \omega_i \mathrm{vec}(\Sigma^{(i)})^{\otimes 2} \in \mathbb{R}^{n_2\times n_2}, \tag{1}$$

$$X_6 = \sum_{i=1}^{k} \omega_i \mathrm{vec}(\Sigma^{(i)})^{\otimes 3} \in \mathbb{R}^{n_2\times n_2\times n_2}. \tag{2}$$

Note that the entries in X_4 and X_6 are quadratic and cubic monomials of the covariance matrices, respectively. If we have X_4 and X_6, the tensor decomposition algorithm in Anandkumar et al. (2014) can be immediately applied to recover ω_i's and $\Sigma^{(i)}$'s under mild conditions. It is easy to verify that those conditions are indeed satisfied with high probability in the smoothed analysis setting.

By Isserlis's theorem, the entries of the moments M_4 and M_6 are indeed quadratic and cubic functions of the covariance matrices, respectively. However, the structure of the true moments M_4 and M_6 have more symmetries, consider for example,

$$[M_4]_{1,2,3,4} = \sum_{i=1}^{k} \omega_i (\Sigma^{(i)}_{1,2}\Sigma^{(i)}_{3,4} + \Sigma^{(i)}_{1,3}\Sigma^{(i)}_{2,4} + \Sigma^{(i)}_{1,4}\Sigma^{(i)}_{2,3}),$$

while $[X_4]_{(1,2),(3,4)} = \sum_{i=1}^{k} \omega_i \Sigma^{(i)}_{1,2}\Sigma^{(i)}_{3,4}$. Note that due to symmetry, the number of distinct entries in M_4 ($\binom{n+3}{4} \approx n^4/24$) is much smaller than the number of distinct entries in X_4 ($\binom{n_2+1}{2} \approx n^4/8$). Similar observation can be made about M_6 and X_6.

[4] Note that by standard random matrix theory, with high probability the 4-th step is feasible and the perturbation procedure in Definition 3.2 succeeds. Also, with high probability we have $\|\widetilde{\mu}^{(i)}\| \leq 1$ and $0 \preceq \widetilde{\Sigma}^{(i)} \preceq I_n$ for all $i \in [k]$.

[5] Note that the algorithms of Belkin and Sinha (2010) and Moitra and Valiant (2010) run in polynomial time for fixed k.

Therefore, it is not immediate how to find the desired X_4 and X_6 based on M_4 and M_6. We call the moments M_4, M_6 the *folded moments* as they have more symmetry, and the corresponding X_4, X_6 the *unfolded moments*. One of the key steps in our algorithm is to unfold the true moments M_4, M_6 to get X_4, X_6 by exploiting special structure of M_4, M_6.

In some cases, it is easier to restrict our attention to the entries in M_4 with indices corresponding to distinct variables. In particular, we define

$$\overline{M}_4 = [[M_4]_{j_1,j_2,j_3,j_4} : 1 \le j_1 < j_2 < j_3 < j_4 \le n] \in \mathbb{R}^{n_4}, \tag{3}$$

where $n_4 = \binom{n}{4}$ is the number of 4-tuples with indices corresponding to distinct variables. We define $\overline{M}_6 \in \mathbb{R}^{n_6}$ similarly where $n_6 = \binom{n}{6}$. We will see that these entries are nice as they are *linear projections* of the desired unfolded moments X_4 and X_6 (Lemma 3.7 below), also such projections satisfy certain "symmetric off-diagonal" properties which are convenient for the proof.

LEMMA 3.7. *For a zero-mean Gaussian mixture model, there exist two fixed and known linear mappings $\mathcal{F}_4 : \mathbb{R}^{n_2 \times n_2} \to \mathbb{R}^{n_4}$ and $\mathcal{F}_6 : \mathbb{R}^{n_2 \times n_2 \times n_2} \to \mathbb{R}^{n_6}$ such that:*

$$\overline{M}_4 = \sqrt{3}\mathcal{F}_4(X_4), \quad \overline{M}_6 = \sqrt{15}\mathcal{F}_6(X_6). \tag{4}$$

Moreover $\mathcal{F}_4$ is a projection from a $\binom{n_2+1}{2}$-dimensional subspace to a n_4-dimensional subspace, and $\mathcal{F}_6$ is a projection from a $\binom{n_2+2}{3}$-dimensional subspace to a n_6-dimensional subspace.

4. ALGORITHM OUTLINE FOR LEARNING MIXTURE OF ZERO-MEAN GAUSSIANS

In this section, we present our algorithm for learning zero-mean Gaussian mixture model. The algorithmic ideas and the analysis are at the core of this paper. Later we show that it is relatively easy to generalize the basic ideas and the techniques to handle the general case.

For simplicity we state our algorithm using the exact moments $\widetilde{M}_4$ and $\widetilde{M}_6$, while in implementation the empirical moments $\widehat{M}_4$ and $\widehat{M}_6$ obtained with the samples are used. In later sections, we verify the correctness of the algorithm and show that it is robust: the algorithm learns the parameters up to arbitrary accuracy using polynomial number of samples.

STEP 1. *Span Finding: Find the span of covariance matrices .*

(a) *For a set of indices $\mathcal{H} \subset [n]$ of size $|\mathcal{H}| = \sqrt{n}$, find the span:*

$$\mathcal{S} = span\left\{\widetilde{\Sigma}^{(i)}_{[:,j]} : i \in [k], j \in \mathcal{H}\right\} \subset \mathbb{R}^n. \tag{5}$$

(b) *Find the span of the covariance matrices with the columns projected onto $\mathcal{S}^\perp$, namely,*

$$\mathcal{U}_S = span\left\{vec(Proj_{S^\perp}\widetilde{\Sigma}^{(i)}) : i \in [k]\right\} \subset \mathbb{R}^{n^2}. \tag{6}$$

(c) *For two disjoint sets of indices $\mathcal{H}_1$ and $\mathcal{H}_2$, repeat Step 1 (a) and Step 1 (b) to obtain $\mathcal{U}_1$ and $\mathcal{U}_2$, namely the span of covariance matrices projected onto two subspaces $\mathcal{S}_1^\perp$ and $\mathcal{S}_2^\perp$. Merge $\mathcal{U}_1$ and $\mathcal{U}_2$ to obtain the span of covariance matrices $\mathcal{U}$:*

$$\mathcal{U} = span\left\{\widetilde{\Sigma}^{(i)} : i \in [k]\right\} \subset \mathbb{R}^{n_2}. \tag{7}$$

STEP 2. *Unfolding: Recover the unfolded moments $\widetilde{X}_4, \widetilde{X}_6$. Given the folded moments $\widetilde{\overline{M}}_4, \widetilde{\overline{M}}_6$ as defined in (3), and given the subspace $U \in \mathbb{R}^{n_2 \times k}$ from Step 1, let $\widetilde{Y}_4 \in \mathbb{R}^{k \times k}_{sym}$ and $\widetilde{Y}_6 \in \mathbb{R}^{k \times k \times k}_{sym}$ be the unknowns, solve the following systems of linear equations.*

$$\widetilde{\overline{M}}_4 = \sqrt{3}\mathcal{F}_4(U\widetilde{Y}_4U^\top), \quad \widetilde{\overline{M}}_6 = \sqrt{15}\mathcal{F}_6(\widetilde{Y}_6(U^\top, U^\top, U^\top)). \tag{8}$$

The unfolded moments $\widetilde{X}_4, \widetilde{X}_6$ are then given by

$$\widetilde{X}_4 = U\widetilde{Y}_4U^\top, \widetilde{X}_6 = \widetilde{Y}_6(U^\top, U^\top, U^\top).$$

STEP 3. *Tensor Decomposition: learn $\widetilde{\omega}_i$ and $\widetilde{\Sigma}^{(i)}$ from $\widetilde{Y}_4$ and $\widetilde{Y}_6$.*
Given U, and given $\widetilde{Y}_4$ and $\widetilde{Y}_6$ which are relate to the parameters as follows:

$$\widetilde{Y}_4 = \sum_{i=1}^k \widetilde{\omega}_i (U^\top \widetilde{\Sigma}^{(i)})^{\otimes 2}, \quad \widetilde{Y}_6 = \sum_{i=1}^k \widetilde{\omega}_i (U^\top \widetilde{\Sigma}^{(i)})^{\otimes 3},$$

we apply tensor decomposition techniques to recover $\widetilde{\Sigma}^{(i)}$'s and $\widetilde{\omega}_i$'s.

5. IMPLEMENTING THE STEPS FOR MIXTURE OF ZERO-MEAN GAUSSIANS

In this part we show how to accomplish each step of the algorithm outlined in Section 4 and sketch the proof ideas.

For each step, we first explain the detailed algorithm, and list the deterministic conditions on the underlying parameters as well as on the *exact* moments for the step to work correctly. Then we show that these deterministic conditions are satisfied with high probability over the ρ-perturbation of the parameters in the smoothed analysis setting. In order to analyze the sample complexity, we further show that when we are given the *empirical* moments which are close to the exact moments, the output of the step is also close to that in the exact case.

In particular we show the correctness and the stability of each step in the algorithm with two main lemmas: the first lemma shows that with high probability over the random perturbation of the covariance matrices, the exact moments satisfy the deterministic conditions that ensure the correctness of each step; the second lemma shows that when the algorithm for each step works correctly, it is actually stable even when the moments are estimated from finite samples and have only inverse polynomial accuracy to the exact moments.

Step 1: Span Finding.

Given the 4-th order moments $\widetilde{M}_4$, Step 1 finds the span of covariance matrices $\mathcal{U}$ as defined in (7). Note that by definition of the unfolded moments $\widetilde{X}_4$ in (1), the subspace $\mathcal{U}$ coincides with the column span of the matrix $\widetilde{X}_4$.

By Lemma 3.7, we know that the entries in $\widetilde{M}_4$ are linear mappings of entries in $\widetilde{X}_4$. Since the matrix $\widetilde{X}_4$ is of low rank

($k \ll n_2$), this corresponds to the *matrix sensing* problem first studied in Recht et al. (2010). In general, matrix sensing problems can be hard even when we have many linear observations (Hardt et al. (2014b)). Previous works (Recht et al. (2010); Hardt et al. (2014a); Jain et al. (2013)) showed that if the linear mapping satisfy *matrix RIP* property, one can uniquely recover $\widetilde{X}_4$ from $\widetilde{M}_4$.

However, properties like RIP do not hold in our setting where the linear mapping is determined by Isserlis' Theorem. We can construct two different mixtures of Gaussians with different unfolded moments $\widetilde{X}_4$, but the same folded moment $\widetilde{M}_4$. Therefore the existing matrix recovery algorithm cannot be applied, and we need to develop new tools by exploiting the special moment structure of Gaussian mixtures.

Step 1 (a). Find the Span of a Subset of Columns of the Covariance Matrices.

The key observation for this step is that if we hit $\widetilde{M}_4$ with three basis vectors, we get a vector that lies in the span of the columns of the covariance matrices:

CLAIM 5.1. *For a mixture of zero-mean Gaussians $\mathcal{G} = \{(\omega_i, 0, \Sigma^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$, the one-dimensional slices of the 4-th order moments M_4 are given by: $\forall j_1, j_2, j_3 \in [n]$*

$$M_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \mathbf{e}_{j_3}, I) = \sum_{i=1}^{k} \omega_i \left(\Sigma^{(i)}_{j_1,j_2}\Sigma^{(i)}_{[:,j_3]} + \Sigma^{(i)}_{j_1,j_3}\Sigma^{(i)}_{[:,j_2]} + \Sigma^{(i)}_{j_2,j_3}\Sigma^{(i)}_{[:,j_1]}\right). \quad (9)$$

In particular, if we pick the indices j_1, j_2, j_3 in the index set $\mathcal{H}$, the vector $M_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \mathbf{e}_{j_3}, I)$ lies in the desired span $\mathcal{S} = \left\{\Sigma^{(i)}_{[:,j]} : i \in [k], j \in \mathcal{H}\right\}$.

We shall partition the set $\mathcal{H}$ into three disjoint subsets $\mathcal{H}^{(i)}$ of equal size $\sqrt{n}/3$, and pick $j_i \in H^{(i)}$ for $i = 1, 2, 3$. In this way, we have $(|\mathcal{H}|/3)^3 = \Omega(n^{1.5})$ such one-dimensional slices of M_4, which all lie in the desired subspace $\mathcal{S}$. Moreover, the dimension of the subspace $\mathcal{S}$ is at most $k|\mathcal{H}| \ll n^{1.5}$. Therefore, with the ρ-perturbed parameters $\widetilde{\Sigma}^{(i)}$'s, we can expect that with high probability the slices of $\widetilde{M}_4$ span the entire subspace $\mathcal{S}$.

CONDITION 5.2 (DETERMINISTIC CONDITION). *Let $\widetilde{Q}_S \in \mathbb{R}^{n\times(|\mathcal{H}|/3)^3}$ be the matrix whose columns are the vectors $\widetilde{M}_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, \mathbf{e}_{j_3}, I)$ for $j_i \in \mathcal{H}^{(i)}$. If the matrix $\widetilde{Q}_S$ achieves its maximal column rank $k|\mathcal{H}|$, we can find the desired span $\mathcal{S}$ defined in (5) by the column span of matrix $\widetilde{Q}_S$.*

We first show that this deterministic condition is satisfied with high probability by bounding the $k|\mathcal{H}|$-th singular value of $\widetilde{Q}_S$ with smoothed analysis.

LEMMA 5.3 (CORRECTNESS). *Given the exact 4-th order moments $\widetilde{M}_4$, for any index set $\mathcal{H}$ of size $|\mathcal{H}| = \sqrt{n}$, With high probability, the $k|\mathcal{H}|$-th singular value of $\widetilde{Q}_S$ is at least $\Omega(\omega_o\rho^2 n)$.*

The proof idea involves writing the matrix $\widetilde{Q}_S$ as a product of three matrices, and using the results on spectral properties of random matrices Rudelson and Vershynin (2009) to show that with high probability the smallest singular value of each factor is lower bounded.

Since this step only involves the singular value decomposition of the matrix $\widetilde{Q}_S$, we then use the standard matrix perturbation theory to show that this step is stable:

LEMMA 5.4 (STABILITY). *Given the empirical estimator of the 4-th order moments $\widehat{M}_4 = \widetilde{M}_4 + E_4$, suppose that the entries of E_4 have absolute value at most δ. Let the columns of matrix $\widetilde{S} \in \mathbb{R}^{n\times k|\mathcal{H}|}$ be the left singular vector of $\widetilde{Q}_S$, and let $\widehat{S}$ be the corresponding matrix obtained with $\widehat{M}_4$. When δ is inverse polynomially small, the distance between the two projections $\|Proj_{\widehat{S}} - Proj_{\widetilde{S}}\|$ is upper bounded by $O\left(n^{1.25}\delta/\sigma_{k|\mathcal{H}|}(\widetilde{Q}_S)\right)$.*

REMARK 5.5. *Note that we need the high dimension assumption ($n \gg k$) to guarantee the correctness of this step: in order to span the subspace $\mathcal{S}$, the number of distinct vectors should be equal or larger than the dimension of the subspace, namely $|\mathcal{H}|^3 \geq k|\mathcal{H}|$; and the subspace should be non-trivial, namely $k|\mathcal{H}| < n$. These two inequalities suggest that we need $n \geq \Omega(k^{1.5})$. However, we used the stronger assumption $n \geq \Omega(k^2)$ to obtain the lower bound of the smallest singular value in the proof.*

Step 1 (b). Find the Span of Projected Covariance Matrices.

In this step, we continue to use the structural properties of the 4-th order moments. In particular, we look at the two-dimensional slices of M_4 obtained by hitting it with two basis vectors:

CLAIM 5.6. *For a mixture of zero-mean Gaussians $\mathcal{G} = \{(\omega_i, 0, \Sigma^{(i)})\}_{i\in[k]} \in \mathcal{G}_{n,k}$, the two-dimensional slices of the 4-th order moments M_4 are given by: $\forall j_1, j_2 \in [n]$,*

$$M_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, I, I) = \sum_{i=1}^{k} \omega_i \left(\Sigma^{(i)}_{j_1,j_2}\Sigma^{(i)} + \Sigma^{(i)}_{[:,j_1]}(\Sigma^{(i)}_{[:,j_2]})^\top + \Sigma^{(i)}_{[:,j_2]}(\Sigma^{(i)}_{[:,j_1]})^\top\right), \quad (10)$$

Note that if we take the indices j_1 and j_2 in the index set $\mathcal{H}$, the slice $M_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, I, I)$ is *almost* in the span of the covariance matrices, except $2k$ additive rank-one terms in the form of $\Sigma^{(i)}_{[:,j_1]}(\Sigma^{(i)}_{[:,j_2]})^\top$. These rank-one terms can be eliminated by projecting the slice to the subspace $\mathcal{S}^\perp$ obtained in Step 1 (a), namely, $\forall j_1, j_2 \in \mathcal{H}$,

$$\mathrm{vec}(\mathrm{Proj}_{S^\perp} M_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, I, I)) = \sum_{i=1}^{k} \omega_i \Sigma^{(i)}_{j_1,j_2} \mathrm{vec}(\mathrm{Proj}_{S^\perp}\Sigma^{(i)}),$$

and this projected two-dimensional slice lies in the desired span $\mathcal{U}_S$ as defined in (6). Moreover, there are $\binom{|\mathcal{H}|+1}{2} = \Omega(n)$ such projected two-dimensional slices, while the dimension of the desired span $\mathcal{U}_S$ is at most k.

CONDITION 5.7 (DETERMINISTIC CONDITION). *Let $\widetilde{Q}_{U_S} \in \mathbb{R}^{n_2\times|\mathcal{H}|(|\mathcal{H}|+1)/2}$ be a matrix whose (j_1, j_2)-th column for is equal to the projected two-dimensional slice*

$$vec(Proj_{S^\perp}\widetilde{M}_4(\mathbf{e}_{j_1}, \mathbf{e}_{j_2}, I, I)),$$

for $j_1 \leq j_2$ and $j_1, j_2 \in \mathcal{H}$. If the matrix $\widetilde{Q}_{U_S}$ achieves its maximal column rank k, the desired span $\mathcal{U}_S$ defined in (6) is given by the column span of the matrix $\widetilde{Q}_{U_S}$.

We show that this deterministic condition is satisfied by bounding the k-th singular value of $\widetilde{Q}_{U_S}$ in the smoothed analysis setting:

LEMMA 5.8 (CORRECTNESS). *Given the exact 4-th order moments $\widetilde{M}_4$, with high probability, the k-th singular value of $\widetilde{Q}_{U_S}$ is at least $\Omega(\omega_o \rho^2 n^{1.5})$.*

Similar to Lemma 5.3, the proof is based on writing the matrix Q_{U_S} as a product of three matrices, then bound their k-th singular values using random matrix theory. The stability analysis also relies on the matrix perturbation theory.

LEMMA 5.9 (STABILITY). *Given the empirical 4-th order moments $\widehat{M}_4 = \widetilde{M}_4 + E_4$, assume that the absolute value of entries of E_4 are at most δ_2. Also, given the output $Proj_{\widehat{S}^\perp}$ from Step 1 (a), and assume that $\|Proj_{\widehat{S}^\perp} - Proj_{\widetilde{S}^\perp}\| \leq \delta_1$. When δ_1 and δ_2 are inverse polynomially small, we have*

$$\|Proj_{\widehat{U}_S} - Proj_{\widetilde{U}_S}\| \leq O\left(n^{2.5}\left(\delta_2 + 2\delta_1\right)/\sigma_k(\widetilde{Q}_{U_S})\right).$$

Step 1 (c). Merge $\mathcal{U}_1$,$\mathcal{U}_2$ to get the span of covariance matrices $\mathcal{U}$.

Note that for a given index set $\mathcal{H}$, the span $\mathcal{U}_S$ obtained in Step 1 (b) only gives partial information about the span of the covariance matrices. The idea of getting the span of the full covariance matrices is to obtain two sets of such partial information and then merge them.

In order to achieve that, we repeat Step 1 (a) and Step 1 (b) for two *disjoint* sets $\mathcal{H}_1$ and $\mathcal{H}_2$, each of size $\sqrt{n}$. The two subspace S_1 and S_2 thus correspond to the span of two disjoint sets of covariance matrix columns. Therefore, we can hope that U_1 and U_2, the span of covariance matrices projected to $S_1^\perp$ and $S_2^\perp$ contain enough information to recover the full span U.

In particular, we prove the following claim:

CONDITION 5.10 (DETERMINISTIC CONDITION). *Let the columns of two (unknown) matrices $V_1 \in \mathbb{R}^{n\times k}$ and $V_2 \in \mathbb{R}^{n\times k}$ form two basis of the same k-dimensional (unknown) subspace $\mathcal{U} \subset \mathbb{R}^n$, and let U denote an arbitrary orthonormal basis of $\mathcal{U}$. Given two s-dimensional subspaces S_1 and S_2, denote $S_3 = S_1^\perp \cup S_2^\perp$. Given two projections of $\mathcal{U}$ onto the two subspaces $S_1^\top$ and $S_2^\top$: $U_1 = Proj_{S_1^\perp} V_1$ and $U_2 = Proj_{S_2^\perp} V_2$. If $\sigma_{2s}([S_1, S_2]) > 0$ and $\sigma_k(Proj_{S_3} U) > 0$, there is an algorithm for finding $\mathcal{U}$ robustly.*

The main idea in the proof is that since s is not too large, the two subspaces $S_1^\perp$ and $S_2^\perp$ have a large intersection. Using this intersection we can "align" the two basis V_1 and V_2 and obtain $V_1^\dagger V_2$, and then it is easy to merge the two projections of the same matrix (instead of a subspace).

Moreover, we show that when applying this result to the projected span of covariance matrices, we have $s = k|\mathcal{H}| \leq n/3$, and the two deterministic conditions $\sigma_{2s}([S_1, S_2]) > 0$ and $\sigma_k(\mathrm{Proj}_{S_3} V_1) > 0$ are indeed satisfied with high probability over the parameter perturbation.

Step 2. Unfold the moments to get $\widetilde{X}_4$ and $\widetilde{X}_6$.

We show that given the span of covariance matrices $\mathcal{U}$ obtained from Step 1, finding the unfolded moments $\widetilde{X}_4$, $\widetilde{X}_6$ is reduced to solving two systems of linear equations.

Recall that the challenge of recovering $\widetilde{X}_4$ and $\widetilde{X}_6$ is that the two linear mappings $\mathcal{F}_4$ and $\mathcal{F}_6$ defined in (4) are *not linearly invertible*. The key idea of this step is to make use of the span $\mathcal{U}$ to *reduce the number of variables*. Note that given the basis $U \in \mathbb{R}^{n_2\times k}$ of the span of the covariance matrices, we can represent each vectorized covariance matrix as $\widetilde{\Sigma}^{(i)} = U\widetilde{\sigma}^{(i)}$. Now Let $\widetilde{Y}_4 \in \mathbb{R}^{k\times k}_{sym}$ and $\widetilde{Y}_4 \in \mathbb{R}^{k\times k\times k}_{sym}$ denote the unfolded moments in this new coordinate system:

$$\widetilde{Y}_4 := \sum_{i=1}^{k} \widetilde{\omega}_i \widetilde{\sigma}^{(i)} \otimes^2, \quad \widetilde{Y}_6 = \sum_{i=1}^{k} \widetilde{\omega}_i \widetilde{\sigma}^{(i)} \otimes^3 .$$

Note that once we know $\widetilde{Y}_4$ and $\widetilde{Y}_6$, the unfolded moments $\widetilde{X}_4$ and $\widetilde{X}_6$ are given by $\widetilde{X}_4 = U\widetilde{Y}_4 U^\top$ and $\widetilde{X}_6 = \widetilde{Y}_6(U^\top, U^\top, U^\top)$. Therefore, after changing the variable, we need to solve the two linear equation systems given in (8) with the variables $\widetilde{Y}_4$ and $\widetilde{Y}_6$.

This change of variable significantly reduces the number of unknown variables. Note that the number of distinct entries in $\widetilde{Y}_4$ and $\widetilde{Y}_6$ are $k_2 = \binom{k+1}{2}$ and $k_3 = \binom{k+2}{3}$, respectively. Since $k_2 \leq n_4$ and $k_3 \leq n_6$, we can expect that the linear mapping from $\widetilde{Y}_4$ to $\widetilde{\widetilde{M}}_4$ and the one from $\widetilde{Y}_6$ to $\widetilde{\widetilde{M}}_6$ are linearly invertible. This argument is formalized below.

CONDITION 5.11 (DETERMINISTIC CONDITION). *Rewrite the two systems of linear equations in (8) in their canonical form and let $\widetilde{H}_4 \in \mathbb{R}^{n_4\times k_2}$ and $\widetilde{H}_6 \in \mathbb{R}^{n_6\times k_3}$ denote the coefficient matrices. We can obtain the unfolded moments $\widetilde{X}_4$ and $\widetilde{X}_6$ if the coefficient matrices have full column rank.*

We show with smoothed analysis that the smallest singular value of the two coefficient matrices are lower bounded with high probability:

LEMMA 5.12 (CORRECTNESS). *With high probability over the parameter random perturbation, the k_2-th singular value of the coefficient matrix $\widetilde{H}_4$ is at least $\Omega(\rho^2 n/k)$, and the k_3-th singular value of the coefficient matrix $\widetilde{H}_6$ is at least $\Omega(\rho^3 (n/k)^{1.5})$.*

To prove this lemma we rewrite the coefficient matrix as product of two matrices and bound their smallest singular values separately. One of the two matrices corresponds to a projection of the Kronecker product $\widetilde{\Sigma} \otimes_{kr} \widetilde{\Sigma}$. In the smoothed analysis setting, this matrix is not necessarily incoherent. In order to provide a lower bound to its smallest singular value, we further apply a carefully designed projection to it, and then we use the concentration bounds for Gaussian chaoses to show that after the projection its columns are incoherent, finally we apply Gershgorin's Theorem to bound the smallest singular value [6].

When implementing this step with the empirical moments, we solve two least squares problems instead of solving the system of linear equations. Again using results in matrix perturbation theory and using the lower bound of the smallest singular values of the two coefficient matrices, we show the stability of the solution to the least squares problems:

[6]Note that the idea of unfolding using system of linear equations also appeared in the work of Jain and Oh (2014). However, in order to show the system of linear equations in their setup is robust, i.e., the coefficient matrix has full rank, they heavily rely on the *incoherence* assumption, which we do not impose in the smoothed analysis setting.

LEMMA 5.13 (STABILITY). *Given the empirical moments $\widehat{M}_4 = \widetilde{M}_4 + E_4$, $\widehat{M}_6 = \widetilde{M}_6 + E_6$, and suppose that the absolute value of entries of E_4 and E_6 are at most δ_1. Let $\widehat{U}$, the output of Step 1, be the estimation for the span of the covariance matrices, and suppose that $\|\widehat{U} - \widetilde{U}\| \leq \delta_2$. Let $\widehat{Y}_4$ and $\widehat{Y}_6$ be the least squares solution respectively. When δ_1 and δ_2 are inverse polynomially small, we have $\|\widetilde{Y}_4 - \widehat{Y}_4\|_F \leq O(\sqrt{n_4}(\delta_1 + \delta_2/\sigma_{min}(\widetilde{H}_4))$ and $\|\widetilde{Y}_6 - \widehat{Y}_6\|_F \leq O(\sqrt{n_6}(\delta_1 + \delta_2/\sigma_{min}(\widetilde{H}_6))$.*

Step 3. Tensor Decomposition.

CLAIM 5.14. *Given $\widetilde{Y}_4$, $\widetilde{Y}_6$ and $\widetilde{U}$, the symmetric tensor decomposition algorithm can correctly and robustly find the mixing weights $\widetilde{\omega}_i$'s and the vectors $\widetilde{\sigma}_i$'s, up to some unknown permutation over $[k]$, with high probability over both the randomized algorithm and the parameter perturbation.*

Proof Sketch for the Zero-mean Case.

Theorem 3.5 follows from the previous smoothed analysis and stability analysis lemmas for each step.

First, exploiting the randomness of parameter perturbation, the smoothed analysis lemmas show that the deterministic conditions, which guarantee the correctness of each step, are satisfied with high probability. Then using concentration bounds of Gaussian variables, we show that with high probability over the random samples, the empirical moments $\widehat{M}_4$ and $\widehat{M}_6$ are entrywise δ-close to the exact moments $\widetilde{M}_4$ and $\widetilde{M}_6$. In order to achieve ϵ accuracy in the parameter estimation, we choose δ to be inverse polynomially small, and therefore the number of samples required will be polynomial in the relevant parameters. The stability lemmas show how the errors propagate only "polynomially" through the steps of the algorithm, which is visualized in Figure 1.

6. ALGORITHM OUTLINE FOR LEARNING MIXTURE OF GENERAL GAUSSIANS

In this section, we briefly discuss the algorithm for learning mixture of *general* Gaussians. Figure 2 shows the inputs and outputs of each step in this algorithm. Many steps share similar ideas to those of the algorithm for the zero-mean case in previous sections.

Step 1. Find $\widetilde{Z}$ and $\widetilde{\Sigma}_o$.

Similar to Step 1 in the zero-mean case, this step makes use of the structure of the 4-th order moments $\widetilde{M}_4$, and is achieved in three small steps:

(a) For a subset $\mathcal{H} \subset [n]$ of size $|\mathcal{H}| = \sqrt{n}$, find the span:

$$\mathcal{S} = \text{span}\left\{\widetilde{\mu}^{(i)}, \widetilde{\Sigma}^{(i)}_{[:,j]} : i \in [k], j \in \mathcal{H}\right\} \subset \mathbb{R}^n. \quad (11)$$

(b) Find the span of the covariance matrices with the columns projected onto $\mathcal{S}^\perp$, namely,

$$\mathcal{U}_S = \text{span}\left\{\text{vec}(\text{Proj}_{S^\perp}\widetilde{\Sigma}^{(i)}) : i \in [k]\right\} \subset \mathbb{R}^{n^2}. \quad (12)$$

(c) For disjoint subsets $\mathcal{H}_1$ and $\mathcal{H}_2$, repeat Step 1 (a) and Step 1 (b) to obtain $\mathcal{U}_1$ and $\mathcal{U}_2$, the span of the covariance matrices projected onto the subspaces $\mathcal{S}_1^\perp$ and $\mathcal{S}_2^\perp$. The intersection of the two subspaces $\mathcal{U}_1$ and $\mathcal{U}_2$ gives the span of the mean vectors

$$\widetilde{Z} = \text{span}\left\{\widetilde{\mu}^{(i)}, i \in [k]\right\}.$$

Merge the two subspaces $\mathcal{U}_1$ and $\mathcal{U}_2$ to obtain the span of the covariance matrices projected to the subspace orthogonal to $\widetilde{Z}$, namely

$$\widetilde{\Sigma}_o = \text{span}\left\{\text{Proj}_{\widetilde{Z}^\perp}\widetilde{\Sigma}^{(i)}\text{Proj}_{\widetilde{Z}^\perp} : i \in [k]\right\}.$$

Step 2. Find the Covariance Matrices in the Subspace $\widetilde{Z}^\perp$ and the Mixing Weights $\widetilde{\omega}_i$'s.

The key observation of this step is that when the samples are projected to the subspace orthogonal to all the mean vectors, they are equivalent to samples from a mixture of zero-mean Gaussians with covariance matrices $\widetilde{\Sigma}^{(i)}_o = \text{Proj}_{\widetilde{Z}^\perp}\widetilde{\Sigma}^{(i)}\text{Proj}_{\widetilde{Z}^\perp}$ and with the same mixing weights $\widetilde{\omega}_i$'s. Therefore, projecting the samples to $\widetilde{Z}^\perp$, the subspace orthogonal to the mean vectors, and use the algorithm for the zero-mean case, we can obtain $\widetilde{\Sigma}^{(i)}_o$'s, the covariance matrices projected to this subspace, as well as the mixing weights $\widetilde{\omega}_i$'s.

Step 3. Find the means.

With simple algebra, this step extracts the projected covariance matrices $\widetilde{\Sigma}^{(i)}_o$'s from the 3-rd order moments $\widetilde{M}_3$, the mixing weights $\widetilde{\omega}_i$ and the projected covariance matrices $\widetilde{\Sigma}^{(i)}_o$'s obtained in Step 2.

Step 4. Find the full covariance matrices.

In Step 2, we obtained $\widetilde{\Sigma}^{(i)}_o$, the covariance matrices projected to the subspace orthogonal to all the means. Note that they are equal to matrices $(\widetilde{\Sigma}^{(i)} + \widetilde{\mu}^{(i)}(\widetilde{\mu}^{(i)})^\top)$ projected to the same subspace. We claim that if we can find the span of these matrices $((\widetilde{\Sigma}^{(i)} + \widetilde{\mu}^{(i)}(\widetilde{\mu}^{(i)})^\top)$'s), we can get each matrix $(\widetilde{\Sigma}^{(i)} + \widetilde{\mu}^{(i)}(\widetilde{\mu}^{(i)})^\top)$, and then subtracting the known rank-one component to find the covariance matrix $\widetilde{\Sigma}^{(i)}$. This is similar to the idea of merging two projections of the same subspace in Step 1 (c) for the zero-mean case.

The idea of finding the desired span is to construct a 4-th order tensor:

$$\widetilde{M}'_4 = \widetilde{M}_4 + 2\sum_{i=1}^{k}\widetilde{\omega}_i(\widetilde{\mu}^{(i)\otimes 4}),$$

which corresponds to the 4-th order moments of a mixture of zero-mean Gaussians with covariance matrices $\widetilde{\Sigma}^{(i)} + \widetilde{\mu}^{(i)}(\widetilde{\mu}^{(i)})^\top$ and the same mixing weights $\widetilde{\omega}_i$'s. Then we can then use Step 1 of the algorithm for the zero-mean case to obtain the span of the new covariance matrices, i.e. $span\{\widetilde{\Sigma}^{(i)} + \widetilde{\mu}^{(i)}(\widetilde{\mu}^{(i)})^\top : i \in [k]\}$.

7. CONCLUSION

In this paper we give the first efficient algorithm for learning mixture of general Gaussians in the smoothed analysis setting. In the algorithm we developed new ways of extracting information from lower-order moment structure. This suggests that although the method of moments often involves solving systems of polynomial equations that are in-

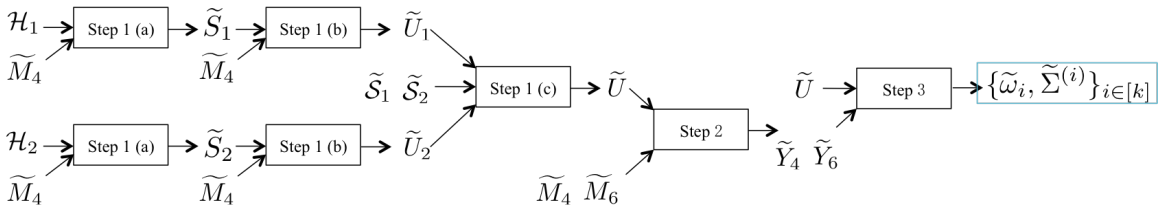

Figure 1: Flow of the algorithm for learning mixture of zero-mean Gaussians.

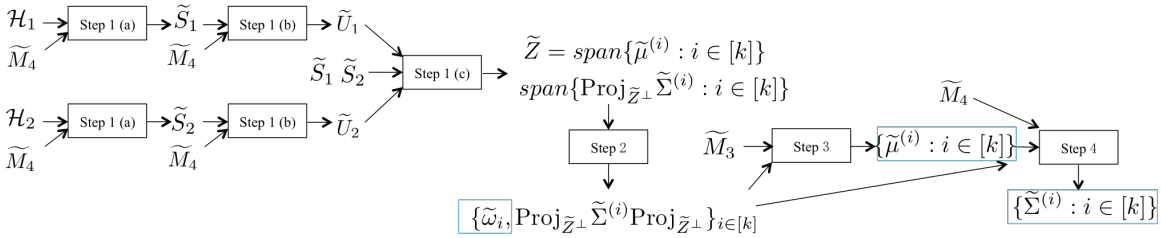

Figure 2: Flow of the algorithm for learning mixtures of general Gaussians.

tractable in general, for natural models there is still hope of utilizing their special structure to obtain algebraic solution.

Smoothed analysis is a very useful way of avoiding degenerate examples in analyzing algorithms. In the analysis, we proved several new results for bounding the smallest singular values of *structured* random matrices. We believe the lemmas and techniques can be useful in more general settings.

Our algorithm uses only up to 6-th order moments. We conjecture that using higher order moments can reduce the number of dimension required to $n \geq \Omega(k^{1+\epsilon})$, or maybe even $n \geq \Omega(k^{\epsilon})$.

Acknowledgements

We thank Santosh Vempala for many insights and for help in earlier attempts at solving this problem, and we thank the anonymous reviewers for their careful reading of our manuscript and their many comments and suggestions which helped us to improve the manuscript.

References

Animashree Anandkumar, Rong Ge, Daniel Hsu, Sham M. Kakade, and Matus Telgarsky. Tensor decompositions for learning latent variable models. *Journal of Machine Learning Research*, 15:2773–2832, 2014. URL `http://jmlr.org/papers/v15/anandkumar14b.html`.

Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James Voss. The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures. *arXiv preprint arXiv:1311.2891*, 2013.

Mikhail Belkin and Kaushik Sinha. Learning gaussian mixtures with arbitrary separation. *arXiv preprint arXiv:0907.1054*, 2009.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 103–112. IEEE, 2010.

Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *Proceedings of the 46th ACM symposium on Theory of computing*, 2014.

S Charles Brubaker and Santosh S Vempala. Isotropic pca and affine-invariant clustering. In *Building Bridges*, pages 241–281. Springer, 2008.

Siu-On Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing*, STOC '14, pages 604–613, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2710-7. doi: 10.1145/2591796.2591848. URL `http://doi.acm.org/10.1145/2591796.2591848`.

Sanjoy Dasgupta. Learning mixtures of gaussians. In *Foundations of Computer Science, 1999. 40th Annual Symposium on*, pages 634–644. IEEE, 1999.

Sanjoy Dasgupta and Leonard J Schulman. A two-round variant of em for gaussian mixtures. In *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pages 152–159. Morgan Kaufmann Publishers Inc., 2000.

Jon Feldman, Rocco A Servedio, and Ryan O'Donnell. Pac learning axis-aligned mixtures of gaussians with no separation assumption. In *Learning Theory*, pages 20–34. Springer, 2006.

Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Proceedings of The 27th Conference on Learning Theory*, pages 703–725, 2014a.

Moritz Hardt, Raghu Meka, Prasad Raghavendra, and Benjamin Weitz. Computational limits for matrix completion. In *Proceedings of The 27th Conference on Learning Theory, COLT 2014, Barcelona, Spain, June 13-15, 2014*, 2014b.

Daniel Hsu and Sham M Kakade. Learning mixtures of spherical gaussians: moment methods and spectral decompositions. In *Proceedings of the 4th conference on In-*

novations in Theoretical Computer Science, pages 11–20. ACM, 2013.

Prateek Jain and Sewoong Oh. Learning mixtures of discrete product distributions using spectral decompositions. In *Proceedings of The 27th Conference on Learning Theory*, pages 824–856, 2014.

Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

Adam Tauman Kalai, Alex Samorodnitsky, and Shang-Hua Teng. Learning and smoothed analysis. In *Foundations of Computer Science, 2009. FOCS'09. 50th Annual IEEE Symposium on*, pages 395–404. IEEE, 2009.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaussians. In *Proceedings of the 42nd ACM symposium on Theory of computing*, pages 553–562. ACM, 2010.

Geoffrey McLachlan and David Peel. *Finite mixture models*. John Wiley & Sons, 2004.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of gaussians. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 93–102. IEEE, 2010.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London. A*, pages 71–110, 1894.

H Permuter, J Francos, and H Jermyn. Gaussian mixture models of texture and colour for image database retrieval. In *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, volume 3, pages III–569. IEEE, 2003.

Benjamin Recht, Maryam Fazel, and Pablo A Parrilo. Guaranteed minimum-rank solutions of linear matrix equations via nuclear norm minimization. *SIAM review*, 52(3):471–501, 2010.

Douglas A Reynolds and Richard C Rose. Robust text-independent speaker identification using gaussian mixture speaker models. *Speech and Audio Processing, IEEE Transactions on*, 3(1):72–83, 1995.

Mark Rudelson and Roman Vershynin. Smallest singular value of a random rectangular matrix. *Communications on Pure and Applied Mathematics*, 62(12):1707–1739, 2009.

Arora Sanjeev and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 247–257. ACM, 2001.

Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms: Why the simplex algorithm usually takes polynomial time. *Journal of the ACM (JACM)*, 51(3): 385–463, 2004.

D Michael Titterington, Adrian FM Smith, Udi E Makov, et al. *Statistical analysis of finite mixture distributions*, volume 7. Wiley New York, 1985.

Gregory John Valiant. *Algorithmic approaches to statistical questions*. PhD thesis, University of California, Berkeley, 2012.

Santosh Vempala and Grant Wang. A spectral algorithm for learning mixture models. *Journal of Computer and System Sciences*, 68(4):841–860, 2004.

Daniel Zoran and Yair Weiss. Natural images, gaussian mixtures and dead leaves. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1736–1744. Curran Associates, Inc., 2012. URL