Proceedings of Machine Learning Research vol 65:1–81, 2017

Robust and Proper Learning for Mixtures of Gaussians via Systems of Polynomial Inequalities

Jerry Li JERRYZLI@MIT.EDU
Ludwig Schmidt LUDWIGS@MIT.EDU
Massachusetts Institute of Technology, Cambridge, MA, 02139, USA

Abstract

Learning a Gaussian mixture model (GMM) is a fundamental statistical problem. One common notion of learning a GMM is proper learning: here, the goal is to find a mixture of k Gaussians $\mathcal{M}$ that is close to the unknown density f from which we draw samples. The distance between $\mathcal{M}$ and f is often measured in the total variation / L_1-distance.

Our main result is an algorithm for learning a mixture of k univariate Gaussians that is *nearly-optimal* for any fixed k. It is well known that the sample complexity of properly learning a univariate k-GMM is $O(k/\epsilon^2)$. However, the best prior *running time* for this problem is $\widetilde{O}(1/\epsilon^{3k-1})$; in particular, the dependence between $1/\epsilon$ and k is exponential. In this paper, we significantly improve this dependence by replacing the $1/\epsilon$ term with $\log 1/\epsilon$, while only increasing the exponent moderately. Specifically, the running time of our algorithm is $(k \cdot \log 1/\epsilon)^{O(k^4)} + \widetilde{O}(k/\epsilon^2)$. For any fixed k, the $\widetilde{O}(k/\epsilon^2)$ term dominates our running time, and thus our algorithm runs in time which is *nearly-linear* in the number of samples drawn. Achieving a running time of poly$(k, 1/\epsilon)$ for proper learning of k-GMMs has recently been stated as an open problem by multiple researchers, and we make progress on this question.

Our main algorithmic ingredient is a new connection between proper learning of parametric distributions and systems of polynomial inequalities. We leverage results for piecewise polynomial approximation of GMMs and reduce the learning problem to a much smaller sub-problem. While tihs sub-problem is still non-convex, its size depends only logarithmically on the final accuracy ϵ. Hence we can invoke computationally expensive methods for solving the sub-problem.

We show that our connection is also useful in the multivariate setting, where we get new results for learning a mixture of two spherical Gaussians. A variant of our approach is also within reach of modern computer algebra systems. Experiments for learning a 2-GMM show promising results: our algorithm improves over the popular Expectation-Maximization (EM) algorithm in the noisy setting.

Keywords: Mixtures of Gaussians, GMM, proper learning, expectation maximization

1. Introduction

Gaussian mixture models (GMMs) are one of the most fundamental probabilistic models. GMMs have a long history in statistics, going back to the seminal work of Pearson (1894). In spite of their age, GMMs are still part of the core machine learning toolkit: the classical Expectation-Maximization (EM) algorithm for GMMs is implemented in many modern machine learning packages such as the Spark's MLlib[1] and Google's Tensorflow.[2] GMMs are routinely employed in diverse applications across science and engineering, for instance recent advances in deep reinforcement learning by Levine et al. (2016). A key strength of GMMs is their ability to describe multi-modal distributions arising from distinct sub-populations.

The wide use of GMMs (and the EM algorithm in particular) motivates the fundamental question: *What provable guarantees can we give for learning a GMM from samples?* Both computational and statistical complexity are important aspects of this question. Classical methods such as maximum likelihood estimation (MLE) yield non-convex problems when applied to GMMs. This non-convexity makes it challenging to design algorithms that combine provably good sample complexity with sub-exponential running time. In this paper, we make progress on the computational question and significantly weaken the exponential dependence in the running time.

1.1. Notions of learning

There are several natural notions of learning a GMM, all of which have been extensively studied in the learning theory community. The known sample and time complexity bounds differ widely for these related problems, and the corresponding algorithmic techniques are also considerably different. We refer the reader to Table 1 at the end of the introduction for an overview. In order of decreasing hardness, the notions of learning are:

Parameter learning. Parameter learning asks to recover the parameters of the unknown GMM (i.e., the means, variances, and mixing weights) up to some given additive error ϵ.

Proper learning. In proper learning, our goal is to find a GMM M' such that the probability density of our hypothesis M' is close to the true unknown density. Common measures of distance are the KL-divergence and the L_1-distance / total variation distance.

Density estimation. Density estimation requires us to find *any* hypothesis $\widehat{h}$ such that the distance between the density of $\widehat{h}$ and the unknown density is small. In particular, $\widehat{h}$ does not need to be a GMM.

Parameter learning is the most desirable guarantee because it allows us to recover the unknown mixture parameters. This is particularly important when the parameters directly correspond to physical quantities that we wish to infer, e.g., in a biological experiment. However, this power comes at a cost: Hardt and Price (2015) show that $\Omega(\frac{1}{\epsilon^{12}})$ samples are already necessary to learn the parameters of a mixture of two univariate Gaussians with accuracy ϵ.[3] Moreover, the sample complexity of parameter learning scales exponentially with the number of components: for a mixture of k univariate Gaussians, the authors also give a sample complexity lower bound of $\Omega(\frac{1}{\epsilon^{6k-2}})$. This exponential

1. https://spark.apache.org/docs/latest/mllib-clustering.html
2. https://github.com/tensorflow/tensorflow/blob/master/tensorflow/contrib/factorization/python/ops/gmm.py
3. This bound is tight, i.e., the paper also gives an algorithm with time and sample complexity $O(\frac{1}{\epsilon^{12}})$.

dependence between the accuracy ϵ and the number of mixture components k quickly becomes prohibitive even for moderate choices of ϵ and k. Note that the exponential complexity is not an issue of non-convexity: the lower bounds are information-theoretic and unconditional.

Tractable learning guarantees. At first sight, this strong lower bound would indicate that polynomial complexity is hopeless for learning GMMs. However, this result stands in stark contrast to the widespread use of GMMs in combination with the EM algorithm. Since the GMMs produced by the EM algorithm are often sufficient in practice, a relaxed learning guarantee is a good explanation for its success. In particular, it is well known that the EM algorithm is a local heuristic for computing the maximum likelihood estimate. Assuming that the EM algorithm (with a sufficiently large number of restarts) comes close to the MLE, it inherits its attractive properties. One such property is approximation in KL-divergence: Given samples from an unknown density f, the EM algorithm produces an estimate $\widehat{h}$ such that the KL-divergence $D_{KL}(\widehat{h}, f)$ is minimized over all mixtures of k Gaussians. This is precisely the proper learning guarantee stated above for the KL-divergence.

Proper learning sidesteps the thorny sample complexity bounds of parameter learning. For instance, Acharya et al. (2014) show that it is possible to properly learn a mixture of k Gaussians in d dimensions with $\widetilde{O}(\frac{dk^9}{\epsilon^4})$ samples under the L_1-distance. This removes the exponential dependence between ϵ and k in the sample complexity (however, their algorithm still requires exponential time). Thus, a plausible explanation for the efficacy of EM with limited samples is that the algorithm learns the unknown GMM in KL-divergence, as opposed to always obtaining an accurate parameter estimate. This motivates a careful study of proper learning, both from a statistical and computational point of view.

In addition to the tractable sample complexity, proper learning naturally enables robust learning guarantees for KL-divergence and L_1-distance. The KL approximation guarantee of the MLE is robust to model-misspecification and applies to *any* unknown density f, not only a mixture of Gaussians. Such a guarantee is also known as (semi-)agnostic learning. Robustness guarantees are particularly relevant from an empirical point of view. In many settings, GMMs are used as a rough (but still useful) model of data that captures its multimodal structure. The true data generating process is often much more complicated than a true GMM. Hence it is important that a learning algorithm is not overly tailored to the specific generative model and instead gracefully adapts to model-misspecification.

Drawbacks of existing results. Considering the positives attributes of the (idealized) EM algorithm, a natural goal is to establish provable bounds for this method. Indeed, this has been the focus of multiple recent papers: Balakrishnan et al. (2014); Daskalakis et al. (2016); Ji Xu (2016); Jin et al. (2016). Unfortunately, there are two main issues with the EM algorithm:

1. Jin et al. (2016) show that the non-convexity of the likelihood objective affects first order methods significantly more than in other estimation problems. In particular, there exist mixtures of k Gaussians for which the EM algorithm requires $\Omega(e^k)$ random initializations. This GMM is even well separated. Unless there is significant progress on provable initializations for EM, the result precludes a polynomial running time.

2. Robustness in KL-divergence has significant disadvantages.[4] For illustration, consider a single outlier point that we move arbitrarily far from the true data points. In order to compensate for the exponentially decaying tails of the Gaussian distribution, the maximum likelihood estimate must assign an entire mixture component to this outlier, even in the case of two mixture components with equal weights and unit variance. While such extreme outliers can often be removed via pre-processing, the KL-robustness issue also manifests itself in much milder noise scenarios. In Figure 1, we see how a small amount of probability mass near the tail of one Gaussian can significantly reduce the solution quality of the EM algorithm.

Prior work in proper learning of GMMs has addressed the KL-robustness issue by focusing on the L_1-distance, see Daskalakis and Kamath (2014) and Acharya et al. (2014). The L_1-distance considers the absolute difference of probability densities, which makes it more robust to outliers near the tails of the distribution. While these results establish good sample complexity bounds, the computational methods are mainly information-theoretic. Due to the non-convexity of the problem, the algorithms resort to a brute-force search over the parameter space. Concretely, the method yields a time complexity of $\widetilde{O}(\frac{1}{\epsilon^{3k-1}})$ for a mixture of k univariate Gaussians. Note that this resembles the prohibitive $\Omega(1/\epsilon^k)$ lower bound for parameter learning a GMM and is much larger than the sample complexity $O(k/\epsilon^2)$. This discrepancy has been raised as an open problem by Moitra (2014) and Diakonikolas (2016). It stands in contrast to parameter learning and density estimation of GMMs, where we have essentially tight upper and lower bounds in the univariate case.

Density estimation also offers robust learning guarantees. However, known results for density estimation of GMMs produce less natural hypotheses such as kernel density estimates or piecewise polynomials, see Devroye and Lugosi (2001) and Acharya et al. (2017). Compared to proper learning, density estimation methods lack the concise, interpretable, and easy to manipulate representation that a GMM offers. Moreover, density estimation with weaker shape constraints such as log-concavity becomes less tractable as the dimensionality increases. Kim and Samworth (2016) show that estimating a log-concave density in d dimensions requires at least $\Omega_d(1/\epsilon^{(d+1)/2})$ samples.[5] The restriction to a parametric model such as GMMs allows significantly better sample complexity in high dimensions.

1.2. Our contributions

We give a family of new algorithms for proper learning of GMMs. As we describe below, the time complexity of our algorithms significantly improves over prior work while maintaining (or improving) its sample complexity and robustness guarantees.

1.2.1. UNIVARIATE CASE

We make significant progress on the aforementioned open problem in the univariate setting. The univariate case has also been studied by Daskalakis and Kamath (2014). Moreover, many algorithms for learning high-dimensional GMMs rely on reductions to one dimension, see Kalai et al. (2010); Moitra and Valiant (2010); Hardt and Price (2015). Hence it is important to understand this case

4. In addition to the robustness issue outlined here, it is also possible to achieve an infinitely large likelihood. We assign a mixture component to a single data point and then reduce the variance of this component to 0. While this behavior is a nuisance, there are practical methods for guarding against degenerate solutions.
5. This exponential dependence is close to tight, see Diakonikolas et al. (2016d).

in greater detail. Later we will show how our univariate techniques are useful in the multivariate setting.

We prove that an exponential dependence between $\frac{1}{\epsilon}$ and k can be avoided. Our algorithm runs in time which is *nearly-optimal* for any fixed k, i.e. nearly-linear in the optimal number of samples. Assuming a small value of k is a natural regime for proper learning of GMMs where we want to summarize a large number of samples (large $1/\epsilon$) with a small but non-trivial number of Gaussian mixture components. Formally, we obtain the following result:

Theorem 1 *Let f be the pdf of an arbitrary unknown distribution, let k be a positive integer, and let $\epsilon > 0$. Let $\mathrm{OPT}_k = \min_{\mathcal{M}} \|f - \mathcal{M}\|_1$ where $\mathcal{M}$ ranges over all k-GMMs. Then there is an algorithm that draws $\widetilde{O}(\frac{k}{\epsilon^2})$ samples from the unknown distribution and with high probability produces a mixture of k Gaussians such that the corresponding pdf $\widehat{h}$ satisfies $\|f - \widehat{h}\|_1 \leq O(\mathrm{OPT}_k) + \epsilon$. Moreover, the algorithm runs in time $\left(k \cdot \log \frac{1}{\epsilon}\right)^{O(k^4)} + \widetilde{O}\left(\frac{k}{\epsilon^2}\right)$.*

We give a semi-agnostic guarantee: if there is GMM that is OPT_k-close, we return a solution that is $O(\mathrm{OPT}_k) + \epsilon$ close. This is a deviation from classical notions in supervised PAC learning as considered by Kearns et al. (1994). There, the goal usually is to output a solution which is $\mathrm{OPT}_k + \epsilon$ close, i.e., the constant in front of OPT_k is 1. However, such a guarantee is typically impossible in distribution learning, e.g., see Chan et al. (2013). Hence the semi-agnostic guarantee is the natural adaptation for our setting.

We remark that we neither optimized the exponent $O(k^4)$, nor the constant in front of OPT_k. Instead, we see our result as a proof of concept that it is possible to (semi-)agnostically and properly learn a mixture of Gaussians in time that is essentially fixed-parameter optimal. This is in contrast to the best prior results that required $\Omega(1/\epsilon^k)$ time.

We achieve this improvement by restricting the non-convex difficulties to a low-dimensional space. In a nutshell, we reduce the problem size to roughly $\log 1/\epsilon$ in a highly non-linear manner. We then invoke algorithms for systems of polynomial inequalities in this smaller problem domain. Solving such polynomial optimization problems is highly expensive, so it is crucial that the size of our polynomial system depends only logarithmically on the number of samples. Avoiding a brute-force search in the original space via this exponential "dimensionality reduction" is a main contribution of our work. As we describe in more detail below, this step relies on recent results in density estimation and approximation theory for mixtures of Gaussians.

In addition to the results for GMMs, our techniques offer a general scheme for converting *improper* learning algorithms to *proper* algorithms. Our approach applies to any parametric family of distributions that are well approximated by a piecewise polynomial. As a result, we can convert purely approximation-theoretic results into proper learning algorithms for other classes of distributions such as mixtures of Laplace or exponential distributions.

1.2.2. MULTIVARIATE CASE

Next, we apply our *univariate* algorithm to the *multivariate* setting. Here, it gives the best known results for properly learning a mixture of two spherical Gaussians with common covariance and weights. The specific case of 2-GMMs has been the subject of several recent works, see Kalai et al. (2010); Daskalakis and Kamath (2014); Balakrishnan et al. (2014); Hardt and Price (2015); Daskalakis et al. (2016); Ji Xu (2016). Similarly, mixtures with shared spherical covariance have been studied previously by Anderson et al. (2014); Acharya et al. (2014); Balakrishnan et al. (2014);

Daskalakis et al. (2016); Ji Xu (2016). The papers of Balakrishnan et al. (2014); Daskalakis et al. (2016); Ji Xu (2016) also consider the setting of equal weights.

Theorem 2 *Let M be a 2-GMM in d dimensions with common spherical covariance matrix and weights. Then there is an algorithm that draws $\widetilde{O}(\frac{d}{\epsilon^6})$ samples from M and with high probability produces a 2-GMM $\widehat{M}$ such that $\|M - \widehat{M}\|_1 \leq \epsilon$. The algorithm runs in time $\widetilde{O}(\frac{d^2}{\epsilon^{7.5}})$.*

The previous best time complexity for our setting is $\widetilde{O}(\frac{d^3}{\epsilon^8})$, which was etablied by Acharya et al. (2014). Our time complexity improves over this result. In particular, our dependence on the dimension d is optimal up to logarithmic factors because $\Omega(d)$ samples (each of which is d-dimensional) are necessary for proper learning in d dimensions; see Acharya et al. (2014).

For simplicity, we have stated our multivariate guarantee in the non-agnostic setting. Building on the results of Diakonikolas et al. (2016a), we can also extend our techniques to the agnostic setting (see Section 10.2).

Conditional results. We propose a plausible (and purely structural) conjecture about projections of Gaussian mixtures. Under this conjecture, our algorithm directly extends to the case of multivariate k-GMMs and separates the exponential dependence between $\frac{1}{\epsilon}$ and k also in higher dimensions. For a Gaussian mixture $\mathcal{M}_\theta$ and a unit vector $\boldsymbol{u}$, we denote the projection of $\mathcal{M}$ onto $\boldsymbol{u}$ with $\mathcal{M}_{\theta\cdot\boldsymbol{u}}$, i.e., we transform each Gaussian component $\mathcal{N}(\boldsymbol{\mu}, \sigma^2\boldsymbol{I})$ to the univariate $\mathcal{N}(\langle\boldsymbol{\mu}, \boldsymbol{u}\rangle, \sigma^2)$. Formally, we propose the following conjecture:

Conjecture 1 *There exists a set of directions $N \subset \mathbb{R}^k$ with cardinality $|N|$ depending only on k such that the following holds for any two spherical k-GMMs $\mathcal{M}_{\theta_1}$ and $\mathcal{M}_{\theta_2}$ in k dimensions: if $\|\mathcal{M}_{\theta_1\cdot\boldsymbol{u}} - \mathcal{M}_{\theta_2\cdot\boldsymbol{u}}\|_1 \leq \epsilon$ for all $\boldsymbol{u} \in N$, then $\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_1 \leq c_k \cdot \epsilon$, where c_k depends only on k.*

This conjecture essentially states that L_1-closeness in a sufficient number of directions implies L_1-closeness in the entire k-dimensional space. As long as $|N|$ and c_k depend only on k , our algorithm naturally generalizes to properly learning multivariate Gaussians and achieves a running time of $\widetilde{O}(\frac{d^2}{\epsilon^5})$ and sample complexity of $\widetilde{O}(\frac{d}{\epsilon^4})$ for any fixed k. An exponential dependence on k is sufficient for these bounds. We conjecture that GMMs satisfy this property due to the smoothness of the Gaussian pdf and give numerical evidence in 3 to 5 dimensions. Our result for two mixture components (Theorem 2) is essentially based on a proof of Conjecture 1 for the case $k = 2$.

1.2.3. A STEP TOWARDS PRACTICE.

While our approach gives new theoretical guarantees, it relies on sophisticated tools for optimizing systems of polynomial inequalities. Unfortunately, these tools are impractical for non-trivial problem sizes. To overcome this hurdle, we propose a variant of our algorithm that only involves systems of polynomial inequalities *without quantifiers*. The algorithm achieves a somewhat weaker learning guarantee, but the resulting running time has only a $(k \cdot \log 1/\epsilon)^{O(k)}$ term, as opposed to the $O(k^4)$ exponent of our algorithm for learning under the L_1-norm. This modification brings our algorithm within reach of modern computer algebra systems. We have implemented our algorithm in Mathematica and investigated its empirical performance for learning a 2-GMM. In the non-agnostic setting, the empirical sample complexity of our algorithm is competitive with the widely-used Expectation Maximization (EM) algorithm. As soon as the 2-GMM is perturbed with a small amount of noise, our estimator demonstrates significantly better learning accuracy (see Figure 1).

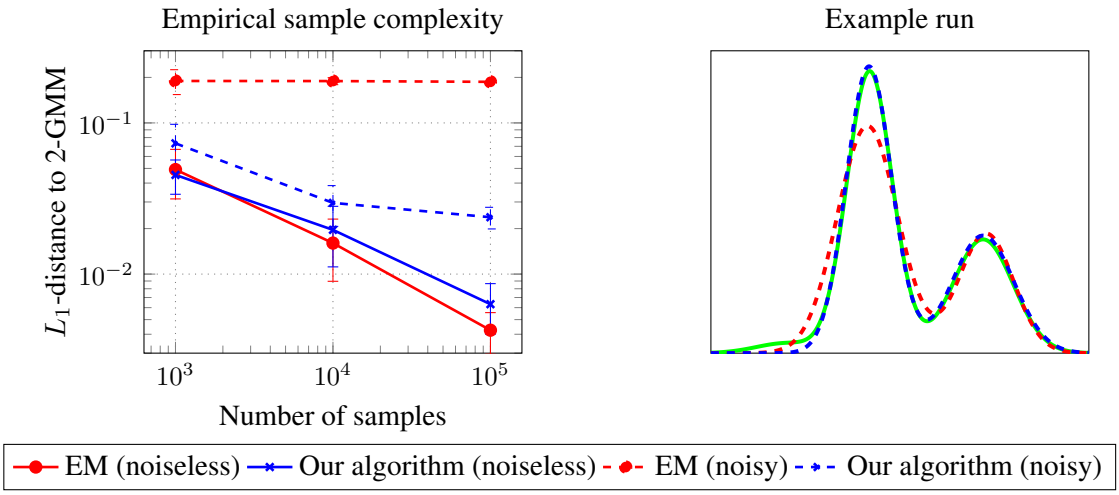

Figure 1: Left plot: empirical sample complexity for learning a 2-GMM in the non-agnostic / noiseless and agnostic / noisy setting. Our algorithm is competitive with Expectation Maximization (EM) in the noiseless case and significantly better when the 2-GMM is perturbed with a small amount of noise (the total noise probability mass is 0.05). Right plot: output of a representative run for our algorithm and the EM algorithm. The green line is the density from which samples are drawn. The slightly heavier left tail significantly affects the accuracy of EM, while our estimator closely matches the true distribution.

We emphasize that our implementation is only a prototype to study the statistical behavior of our polynomial programs. There are many approaches for improving the empirical running time of our algorithm. Experiments with different solution heuristics in Mathematica indicate that local search methods should perform well when solving our systems of polynomial inequalities. The L_2-formulation enables us to run gradient descent with the density approximation objective instead of the likelihood objective of the EM algorithm. Moreover, we can apply gradient descent to a piecewise polynomial approximation that is significantly smaller than the number of samples. Due to the length of the current paper, we defer a more thorough experimental evaluation with several mixture components and multiple dimensions to future work.

On the theoretical side, we prove the following result about our simplified system of polynomial inequalities.

Theorem 3 *Let f be the unknown pdf, and let θ be so that $\|f - \mathcal{M}_\theta\|_1 = \mathrm{OPT}_k$. Let p_{dens} be supported on $[-1,1]$ so that $\|p_{\mathrm{dens}} - \mathcal{M}_\theta\|_1 < O(\mathrm{OPT}_k + \epsilon)$ and $\sup_{x\in\mathbb{R}} |p_{\mathrm{dens}}(x) - \mathcal{M}_\theta(x)| \leq O(\mathrm{OPT}_k + \xi)$. Finally, let $\tau_{\max}^2 = \max_{i=1}^k \tau_i^2$, where $\tau_1, \ldots, \tau_k$ are the precisions for the components of $\mathcal{M}_\theta$. Then there is an algorithm that outputs a k-GMM $\mathcal{M}_{\widehat{\theta}}$ so that with probability $1-\delta$, we have*

$$\|f - \mathcal{M}_{\widehat{\theta}}\|_1 \;\leq\; O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \tau_{\max}\epsilon + \xi)} + \mathrm{OPT}_k + \epsilon\right) .$$

Moreover, the algorithms runs in time $\widetilde{O}\left(\frac{k+\log 1/\delta}{\epsilon^2}\right) + (k \log 1/\epsilon)^{O(k)}$.

When $\tau_{\max}$ and ξ are both reasonable (i.e., $\tau_{\max} = O(1)$ and $\xi = O(\epsilon)$), the error guarantee simplifies to $\|f - \mathcal{M}_{\widehat{\theta}}\|_1 \leq O(\mathrm{OPT}_k) + \epsilon$, which matches our algorithm for the univariate case in Theorem 1.

1.3. Techniques

At its core, our univariate algorithm fits a mixture of Gaussians to a density estimate. We first invoke the algorithm of Acharya et al. (2017) to obtain an "improper" but ϵ-accurate and agnostic density estimate. The time and sample complexity of this step is $\widetilde{O}(\frac{k}{\epsilon^2})$. The resulting density estimate has the form of a piecewise polynomial with $O(k)$ pieces, each of which has degree $O(\log \frac{1}{\epsilon})$. Our algorithm does not draw any further samples after obtaining the density estimate. The process of fitting a mixture of univariate Gaussians is entirely deterministic.

Once we have obtained a good density estimate, we need to approximate it with a mixture of k Gaussians. We reduce this problem to solving a system of polynomial inequalities, for which we employ Renegar's algorithm Renegar (1992a,b). For the univariate case, this reduction to a system of polynomial inequalities is our main technical contribution and relies on the following techniques.

Shape-restricted polynomials. Directly fitting a mixture of Gaussians to the density estimate is challenging because the Gaussian pdf is not convex in the mean and variance parameters. Instead, we utilize *shape restricted* polynomials. We say that a polynomial is shape restricted if its coefficients are in a given semialgebraic set, i.e., a set defined by a finite number of polynomial equalities and inequalities. It is well-known in approximation theory that a single Gaussian can be approximated by a piecewise polynomial consisting of three pieces with degree at most $O(\log \frac{1}{\epsilon})$, e.g., see Timan (1963). So instead of fitting a mixture of k Gaussian, we fit a mixture of k shape-restricted piecewise polynomials. By encoding that the shape-restricted polynomials must be close to Gaussian pdfs, we ensure that the resulting mixture of shape-restricted piecewise polynomials is close to a true mixture of k-Gaussians. After solving the system of polynomial inequalities, it is easy to convert the shape-restricted polynomials back to a proper GMM.

$\mathcal{A}_K$-distance. In our final guarantee for proper learning, we are interested in an approximation in the L_1-norm. However, directly encoding the L_1-norm in the system of polynomial inequalities requires knowledge of the intersections between the density estimate and the mixture of piecewise polynomials (this is necessary to compute the integral of their difference). Since our shape-restricted polynomials can have up to $k \cdot \log \frac{1}{\epsilon}$ crossings, directly using the L_1-norm would lead to an exponential dependence on $\log \frac{1}{\epsilon}$ in our system of polynomial inequalities. Instead, we minimize the closely related $\mathcal{A}_K$-norm from VC (Vapnik–Chervonenkis) theory, see Devroye and Lugosi (2001). For functions with at most $K - 1$ sign changes, the $\mathcal{A}_K$-norm exactly matches the L_1-norm. Since two k-GMMs have at most $O(k)$ intersections, we can replace the L_1-norm with the $\mathcal{A}_K$-norm for $K = O(k)$. In contrast to the L_1-norm, we can encode the $\mathcal{A}_K$-norm with a significantly smaller system of polynomial inequalities.

Adaptively rescaling the density estimate. In order to fit a GMM with Renegar's algorithm, we have to solve our system of polynomial inequalities to sufficiently high accuracy. While Renegar's algorithm has a good dependence on the accuracy parameter, our goal is to give an algorithm for proper learning without *any* assumptions on the GMM. We overcome this technical challenge by adaptively rescaling the parametrization used in our system of polynomial inequalities based on the lengths of the intervals that define the piecewise polynomial density estimate p_{dens}. Since p_{dens}

can only be large on short intervals, the best Gaussian fit to p_{dens} can only have large parameters near such intervals. This allows us to identify where we require more accuracy when computing the mixture parameters.

Reducing multivariate to univariate. The time complexity of our approach based on systems of polynomial inequalities is inherently exponential in the number of GMM parameters. Since the component means are now d-dimensional, naively applying our univariate scheme would yield a running time exponential in d, even if a d-dimensional density estimate was available. Moreover, there are no known algorithms that improperly learn a GMM in d dimensions for $d > 1$. So there is no natural "reference density" for the system of polynomial inequalities. Our algorithm for the multivariate case overcomes these challenges via two reductions.

First, we reduce the d-dimensional learning problem to a k-dimensional problem by finding a subspace close to the subspace spanned by the true component means. This approach has been employed in prior work Acharya et al. (2014), but we simplify the algorithm and improve its running time. In particular, we build on Musco and Musco (2015) and show that it suffices to find an approximate PCA of the covariance matrix.

Second (now for $k = 2$), we reduce the 2-dimensional problem to simultaneously satisfying a set of 1-dimensional contraints. We use a constant-size net in the 2-dimensional space and produce a density estimate for each direction in the net. Then we construct a single system of polynomial inequalities that enforces closeness in all directions of the net. This reduction relies on a structural result about projections of GMMs, showing that univariate L_1-closeness in each direction of the net implies L_1-closeness of the resulting GMM in all of $\mathbb{R}^d$.

1.4. Related work

Due to space constraints, it is impossible to summarize the entire body of work on learning GMMs here. Therefore, we limit our attention to the notions of learning outlined in Subsection 1.1. This is only one part of the picture: as mentioned above, the well-known Expectation-Maximization (EM) algorithm is still the subject of current research, e.g., Balakrishnan et al. (2014); Daskalakis et al. (2016); Ji Xu (2016); Jin et al. (2016).

For parameter learning, the seminal work of Dasgupta Dasgupta (1999) started a long line of research in the theoretical computer science community, e.g., Arora and Kannan (2001); Vempala and Wang (2004); Achlioptas and McSherry (2005); Kannan et al. (2008); Brubaker and Vempala (2008); Brubaker (2009); Kalai et al. (2010); Moitra and Valiant (2010). We refer the reader to Moitra and Valiant (2010) for a discussion of these and related results. Moitra and Valiant (2010) and Belkin and Sinha (2010) were the first to give algorithms that are polynomial in ϵ and the dimension of the mixture while requiring only minimal assumptions on the GMMs. More recently, Hardt and Price (2015) gave tight bounds for learning the parameters of a mixture of two univariate Gaussians: $\Theta(\frac{1}{\epsilon^{12}})$ samples are necessary and sufficient, and the time complexity is linear in the number of samples. Moreover, Hardt and Price give a strong lower bound of $\Omega(\frac{1}{\epsilon^{6k-2}})$ for the sample complexity of parameter learning a k-GMM. While our proper learning algorithm offers a weaker guarantee than these parameter learning approaches, our time and sample complexity avoids the exponential dependence between $\frac{1}{\epsilon}$ and k. See Subsection 1.1 for a discussion regarding parameter and proper learning.

Interestingly, parameter learning becomes more tractable as the number of dimensions increases. A recent line of work investigates this phenomenon under a variety of non-degeneracy assumptions

(e.g., a full-rank matrix of means or smoothed analysis) Hsu and Kakade (2013); Bhaskara et al. (2014); Anderson et al. (2014); Ge et al. (2015). These algorithms require a lower bound on the dimension d such as $d \geq \Omega(k)$ or $d \geq \Omega(k^2)$. Consequently they are not comparable with our result for univariate GMMs. In the multivariate setting, the work closest to ours is Hsu and Kakade (2013), which studies parameter learning of spherical Gaussians with a dependence on the condition number of the component means. For the $k = 2$ case considered in our Theorem 2, the non-degeneracy assumption of Hsu and Kakade (2013) precludes configurations such as two component means on a line. In contrast, our algorithm succeeds for any configuration of the component means and its time and sample complexities do not depend on a condition number.

Proper learning of k-GMMs without separation assumptions was first considered by Feldman et al. (2006), building on work for properly learning mixtures of discrete product distributions in Feldman et al. (2008); Freund and Mansour (1999). For fixed k, their algorithm takes $\mathrm{poly}(d, \frac{1}{\epsilon}, L)$ samples and returns a mixture whose KL-divergence to the unknown mixture is at most ϵ. However, their algorithm has a pseudo-polynomial dependence on L, which is a bound on the means and variances of the underlying components. Such an assumption is not necessary a priori, and our algorithm works without similar requirements. Moreover, their sample and time complexities have an exponential dependence between $\frac{1}{\epsilon}$ and k.

The work closest to ours are the papers Daskalakis and Kamath (2014) and Acharya et al. (2014), who also consider the problem of properly learning a k-GMM. Their algorithms are based on constructing a set of candidate GMMs that are then compared via an improved version of the Scheffé-estimate. While this approach leads to a nearly-optimal sample complexity of $\widetilde{O}(\frac{k}{\epsilon^2})$, their algorithm constructs an exponentially large number of candidate hypothesis. This leads to a time complexity of $O(\frac{1}{\epsilon^{3k-1}})$. As pointed out in Subsection 1.2, our algorithm significantly improves the dependence between $\frac{1}{\epsilon}$ and k.

Recent work of Diakonikolas et al. (2016a) gives new agnostic algorithms for learning high-dimensional spherical GMMs. Our robust high-dimensional algorithm builds upon this work. However, their algorithm ultimately resorts to a brute force search over k dimensions and therefore still runs in time $O(\frac{1}{\epsilon^{3k-1}})$. Our main contribution in the high-dimensional setting is to propose a new algorithmic framework that avoids this brute force search.

Diakonikolas et al. (2016c) provide lower bounds for learning high-dimensional GMMs with statistical query (SQ) algorithms. Their lower bound has an exponential dependence between the dimension d and k, but relies on highly non-spherical Gaussians. Hence their results do not apply to our setting where we study spherical GMMs.

Another related paper is Bhaskara et al. (2015). Their approach reduces the GMM learning problem to finding a sparse solution to a non-negative linear system. Conceptually, this approach is somewhat similar to ours in that they also fit a mixture of Gaussians to a set of density estimates. However, their algorithm does not give a proper learning guarantee: instead of k mixture components, the GMM returned by their algorithm contains $O(\frac{k}{\epsilon^3})$ components. Note that this number of components is significantly larger than the k components returned by our algorithm and increases as the accuracy paramter ϵ improves. In the univariate case, the time and sample complexity of their algorithm is $O(\frac{k}{\epsilon^6})$. Hence their sample complexity is not optimal and roughly $\frac{1}{\epsilon^4}$ worse than our approach. For any fixed k, our running time is also better by roughly $\frac{1}{\epsilon^4}$. In the multivariate setting, both their time and sample complexity is roughly $O((\frac{kd}{\epsilon^3})^d)$, which is *exponential* in the dimension d.

There is a recent line of work on density estimation of structured distributions including GMMs, see Chan et al. (2013, 2014); Acharya et al. (2017). While Acharya et al. (2017) achieves a nearly-optimal time and sample complexity for univariate density estimation of k-GMMs, the hypothesis produced by their algorithm is still a piecewise polynomial. As mentioned in Subsection 1.1, proper learning has multiple advantages over density estimation.

In the context of learning Poisson binomial distributions, subsequent work of Diakonikolas et al. (2016b) has independently used systems of polynomial inequalities for proper learning.[6]

1.5. Outline of our paper

Our paper is divided into three parts.

Univariate algorithm. The first part (Sections 2 to 4) addresses the univariate case, for which we give a proper learning algorithm that is nearly optimal for any fixed number of components. In Section 2, we introduce basic notation and important known results that we utilize in our algorithm. Section 3 describes our univariate learning algorithm for the special case of *well-behaved* density estimates. This assumption allows us to introduce two of our main tools (shape-restricted polynomials and the $\mathcal{A}_K$-distance as a proxy for L_1) without the technical details of adaptively reparametrizing the shape-restricted polynomials. Section 4 then removes this assumption and gives an algorithm that works for agnostically learning *any* mixture of univariate Gaussians. We also show how our techniques can be extended to properly learn further classes of univariate distributions.

Multivariate algorithm. The second part (Sections 5 to 11) extends our univariate algorithm to proper learning of multivariate Gaussian mixtures. In Section 5, we introduce additional preliminaries for the multivariate setting and formally define our multivariate algorithm. Sections 6 to 9 establish the individual building blocks of our algorithm, which we then put together in Section 10. Finally, Section 11 gives numerical evidence for our Conjecture 1.

Experimental algorithm. In the third part (Sections 12 and 13), we give a variant of our univariate algorithm that avoids heavy machinery for systems of polynomial equalities with quantifiers. Section 12 formally defines the algorithm and proves a slightly weaker learning guarantee. We conclude this part with an experimental evaluation of our algorithm in Section 13.

6. To avoid confusion, we remark that the first version of our paper appeared on arXiv on 3 June 2015, see https://arxiv.org/abs/1506.01367.

Problem type	Sample complexity lower bound	Sample complexity upper bound	Time complexity upper bound
Parameter learning			
$k = 2$	$\Theta(\frac{1}{\epsilon^{12}})$ Hardt and Price (2015)	$O(\frac{1}{\epsilon^{12}})$ Hardt and Price (2015)	$O(\frac{1}{\epsilon^{12}})$ Hardt and Price (2015)
general k	$\Omega(\frac{1}{\epsilon^{6k-2}})$ Hardt and Price (2015)	$O((\frac{1}{\epsilon})^{c_k})$ Moitra and Valiant (2010)	$O((\frac{1}{\epsilon})^{c_k})$ Moitra and Valiant (2010)
Proper learning			
$k = 2$	$\Theta(\frac{1}{\epsilon^2})$	$\widetilde{O}(\frac{1}{\epsilon^2})$ Daskalakis and Kamath (2014)	$\widetilde{O}(\frac{1}{\epsilon^5})$ Daskalakis and Kamath (2014)
general k	$\Theta(\frac{k}{\epsilon^2})$	$\widetilde{O}(\frac{k}{\epsilon^2})$ Acharya et al. (2014)	$\widetilde{O}(\frac{1}{\epsilon^{3k-1}})$ Daskalakis and Kamath (2014) Acharya et al. (2014)
Our results			
$k = 2$		$\widetilde{O}(\frac{1}{\epsilon^2})$	$\boldsymbol{\widetilde{O}(\frac{1}{\epsilon^2})}$
general k		$\widetilde{O}(\frac{k}{\epsilon^2})$	$\boldsymbol{(k \log \frac{1}{\epsilon})^{O(k^4)} + \widetilde{O}(\frac{k}{\epsilon^2})}$
Density estimation			
general k	$\Theta(\frac{k}{\epsilon^2})$	$\widetilde{O}(\frac{k}{\epsilon^2})$ Acharya et al. (2017)	$\widetilde{O}(\frac{k}{\epsilon^2})$ Acharya et al. (2017)

Table 1: Overview of the best known results for learning a mixture of univariate Gaussians. Our contributions (highlighted as bold) significantly improve on the previous results for proper learning: the time complexity of our algorithm is nearly optimal for any fixed k. The constant c_k in the time and sample complexity of Moitra and Valiant (2010) depends only on k and is at least k. The sample complexity lower bounds for proper learning and density estimation are folklore results. The only time complexity lower bounds known are the corresponding sample complexity lower bounds.

2. Preliminaries

Before we construct our learning algorithm for GMMs, we introduce basic notation and the necessary tools from density estimation, systems of polynomial inequalities, and approximation theory.

2.1. Basic notation and definitions

For a positive integer k, we write $[k]$ for the set $\{1, \dots, k\}$. Let $I = [\alpha, \beta]$ be an interval. Then we denote the length of I with $|I| = \beta - \alpha$. For a measurable function $f : \mathbb{R} \to \mathbb{R}$, the L_1-norm of f is $\|f\|_1 = \int f(x)\,\mathrm{d}x$. All functions in this paper are measurable.

Since we work with systems of polynomial inequalities, it will be convenient for us to parametrize the normal distribution with the *precision*, i.e., one over the standard deviation, instead of the variance. Thus, throughout the paper we let

$$\mathcal{N}_{\mu,\tau}(x) \stackrel{\text{def}}{=} \frac{\tau}{\sqrt{2\pi}} e^{-\tau^2(x-\mu)^2/2}$$

denote the pdf of a normal distribution with mean μ and precision τ. A k-GMM is a distribution with pdf of the form $\sum_{i=1}^k w_i \cdot \mathcal{N}_{\mu_i,\tau_i}(x)$, where we call the w_i *mixing weights* and require that the w_i satisfy $w_i \geq 0$ and $\sum_{i=1}^k w_i = 1$. Thus a k-GMM is parametrized by $3k$ parameters; namely, the mixing weights, means, and precisions of each component.[7] We let $\Theta_k = \mathcal{S}_k \times \mathbb{R}^k \times \mathbb{R}_+^k$ be the set of parameters, where $\mathcal{S}_k$ is the simplex in k dimensions. For each $\theta \in \Theta_k$, we identify it canonically with $\theta = (w, \mu, \tau)$ where w, μ, and τ are each vectors of length k, and we let

$$\mathcal{M}_\theta(x) = \sum_{i=1}^k w_i \cdot \mathcal{N}_{\mu_i,\tau_i}(x)$$

be the pdf of the k-GMM with parameters θ.

2.2. Important tools

We now turn our attention to results from prior work.

2.2.1. DENSITY ESTIMATION WITH PIECEWISE POLYNOMIALS

Our algorithm uses the following result about density estimation of k-GMMs as a subroutine.

Fact 4 (Acharya et al. (2017)) *Let $k \geq 1$, $\epsilon > 0$ and $\delta > 0$. There is an algorithm* ESTIMATE-DENSITY(k, ϵ, δ) *that satisfies the following properties: the algorithm*

- *takes $\widetilde{O}((k + \log(1/\delta))/\epsilon^2)$ samples from the unknown distribution with pdf f,*
- *runs in time $\widetilde{O}((k + \log 1/\delta)/\epsilon^2)$, and*
- *returns p_{dens}, an $O(k)$-piecewise polynomial of degree $O(\log(1/\epsilon))$ such that*

$$\|f - p_{\mathrm{dens}}\|_1 \leq 4 \cdot \mathrm{OPT}_k + \epsilon$$

with probability at least $1 - \delta$, where

$$\mathrm{OPT}_k = \min_{\theta \in \Theta_k} \|f - \mathcal{M}_\theta\|_1 .$$

7. Note that there are only $3k - 1$ degrees of freedom since the mixing weights must sum to 1.

2.2.2. SYSTEMS OF POLYNOMIAL INEQUALITIES

In order to fit a k-GMM to the density estimate, we solve a carefully constructed system of polynomial inequalities. Formally, a system of polynomial inequalities is an expression of the form

$$S = (Q_1 x^{(1)} \in \mathbb{R}^{n_1}) \dots (Q_v x^{(v)} \in \mathbb{R}^{n_v}) P(y, x^{(1)}, \dots, x^{(v)})$$

where

- the $y = (y_1 \dots, y_\ell)$ are free variables,
- for all $i \in [v]$, the quantifier Q_i is either $\exists$ or $\forall$,
- $P(y, x^{(1)}, \dots, x^{(v)})$ is a quantifier-free Boolean formula with m predicates of the form

  $$g_i(y, x^{(1)}, \dots, x^{(v)}) \ \Delta_i \ 0$$

  where each g_i is a real polynomial of degree d, and where the relations Δ_i are of the form $\Delta_i \in \{>, \geq, =, \neq, \leq, <\}$. We call such predicates *polynomial predicates*.

We say that $y \in \mathbb{R}^\ell$ is a λ-*approximate solution* for this system of polynomial inequalities if there exists a $y' \in \mathbb{R}^\ell$ such that y' satisfies the system and $\|y - y'\|_2 \leq \lambda$. We use the following result by Renegar as a black-box:

Fact 5 (Renegar (1992a,b)) *Let $0 < \lambda < \eta$ and let S be a system of polynomial inequalities as defined above. Then there is an algorithm* SOLVE-POLY-SYSTEM(S, λ, η) *that finds a λ-approximate solution if there exists a solution y with $\|y\|_2 \leq \eta$. If no such solution exists, the algorithm returns "NO-SOLUTION". In any case, the algorithm runs in time*

$$(md)^{2^{O(v)}\ell} \prod_k n_k \log\log\left(3 + \frac{\eta}{\lambda}\right) .$$

2.2.3. SHAPE-RESTRICTED POLYNOMIALS

Instead of fitting Gaussian pdfs to our density estimate directly, we work with piecewise polynomials as a proxy. Hence we need a good approximation of the Gaussian pdf with a piecewise polynomial. In order to achieve this, we use three pieces: two flat pieces that are constant 0 for the tails of the Gaussian, and a center piece that is given by the Taylor approximation.

Let let $T_d(x)$ be the degree-d Taylor series approximation to $\mathcal{N}$ around zero. It is straightforward to show:

Lemma 6 *Let $\epsilon, K > 0$ and let $T_d(x)$ denote the degree-d Taylor expansion of the Gaussian pdf $\mathcal{N}$ around 0. For $d = 2K \log(1/\epsilon)$, we have*

$$\int_{2\sqrt{\log 1/\epsilon}}^{2\sqrt{\log 1/\epsilon}} |\mathcal{N}(x) - T_d(x)| \, \mathrm{d}x \ \leq \ O\left(\epsilon^K \sqrt{\log(1/\epsilon)}\right) .$$

Definition 7 (Shape-restricted polynomials) *Let K be such that*

$$\int_{-2\sqrt{\log 1/\epsilon}}^{2\sqrt{\log 1/\epsilon}} |\mathcal{N}(x) - T_{2K\log(1/\epsilon)}(x)| \, \mathrm{d}x \ < \ \frac{\epsilon}{4} .$$

From Lemma 6 we know that such a K always exists. For any $\epsilon > 0$, let $\widetilde{\mathcal{P}}_\epsilon(x)$ denote the piecewise polynomial function defined as follows:

$$\widetilde{\mathcal{P}}_\epsilon(x) \;=\; \begin{cases} T_{2K\log(1/\epsilon)}(x) & \text{if } x \in [-2\sqrt{\log(1/\epsilon)},\, 2\sqrt{\log(1/\epsilon)}] \\ 0 & \text{otherwise} \end{cases} .$$

For any set of parameters $\theta \in \Theta_k$, let

$$P_{\epsilon,\theta}(x) \;=\; \sum_{i=1}^{k} w_i \cdot \tau_i \cdot \widetilde{\mathcal{P}}_\epsilon(\tau_i(x-\mu_i)) \,.$$

It is important to note that $P_{\epsilon,\theta}(x)$ is a polynomial *both* as a function of θ and as a function of x. This allows us to fit such shape-restricted polynomials with a system of polynomial inequalities. Moreover, our shape-restricted polynomials are good approximations to GMMs. By construction, we get the following result:

Lemma 8 *Let $\theta \in \Theta_k$. Then $\|\mathcal{M}_\theta - P_{\epsilon,\theta}\|_1 \le \epsilon$.*

Proof We have

$$\begin{aligned} \|\mathcal{M}_\theta - P_{\epsilon,\theta}\|_1 &= \int |\mathcal{M}_\theta(x) - P_{\epsilon,\theta}(x)|\,\mathrm{d}x \\ &\overset{(a)}{\le} \sum_{i=1}^{k} w_i \int |\tau_i \cdot \mathcal{N}(\tau_i(x-\mu_i)) - \tau_i \cdot \widetilde{\mathcal{P}}_\epsilon(\tau_i(x-\mu_i))|\,\mathrm{d}x \\ &\overset{(b)}{\le} \sum_{i=1}^{k} w_i \cdot \|\mathcal{N} - \widetilde{\mathcal{P}}_\epsilon\|_1 \\ &\overset{(c)}{\le} \sum_{i=1}^{k} w_i \cdot \epsilon \\ &\le \epsilon \,. \end{aligned}$$

Here, (a) follows from the triangle inequality, (b) from a change of variables, and (c) from the definition of $\widetilde{\mathcal{P}}_\epsilon$. ■

2.2.4. $\mathcal{A}_K$-NORM AND INTERSECTIONS OF k-GMMS

In our system of polynomial inequalities, we must encode the constraint that the shape-restricted polynomials are a good fit to the density estimate. For this, the following notion of distance between two densities will become useful.

Definition 9 ($\mathcal{A}_K$-norm) *Let $\mathfrak{I}_K$ denote the family of all sets of K disjoint intervals $\mathcal{I} = \{I_1, \ldots, I_K\}$. For any measurable function $f : \mathbb{R} \to \mathbb{R}$, we define the $\mathcal{A}_K$-*norm *of f to be*

$$\|f\|_{\mathcal{A}_K} \overset{def}{=} \sup_{\mathcal{I} \in \mathfrak{I}_K} \sum_{I \in \mathcal{I}} \left| \int_I f(x)\,\mathrm{d}x \right| .$$

For functions with few zero-crossings, the $\mathcal{A}_K$-norm is close to the L_1-norm. More formally, we have the following properties, which are easy to check:

Lemma 10 *Let $f : \mathbb{R} \to \mathbb{R}$ be a real function. Then for any $K \geq 1$, we have*

$$\|f\|_{\mathcal{A}_K} \leq \|f\|_1 .$$

Moreover, if f is continuous and there are at most $K-1$ distinct values x for which $f(x) = 0$, then

$$\|f\|_{\mathcal{A}_K} = \|f\|_1 .$$

The second property makes the $\mathcal{A}_K$-norm useful for us because linear combinations of Gaussians have few zeros.

Fact 11 (Kalai et al. (2010) Proposition 7) *Let f be a linear combination of k Gaussian pdfs with variances $\sigma_1, \ldots, \sigma_k$ so that $\sigma_i \neq \sigma_j$ for all $i \neq j$. Then there are at most $2(k-1)$ distinct values x such that $f(x) = 0$.*

These facts give the following corollary.

Corollary 12 *Let $\theta_1, \theta_2 \in \Theta_k$ and let $K \geq 4k$. Then*

$$\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_{\mathcal{A}_K} = \|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_1 .$$

Proof For any $\gamma > 0$, let $\theta_1^\gamma, \theta_2^\gamma$ be so that $\|\theta_i^\gamma - \theta_i\|_\infty \leq \gamma$ for $i \in \{1, 2\}$, and so that the variances of all the components in $\theta_1^\gamma, \theta_2^\gamma$ are all distinct. Lemma 10 and Fact 11 together imply that $\|M_{\theta_1^\gamma} - M_{\theta_2^\gamma}\|_1 = \|M_{\theta_1^\gamma} - M_{\theta_2^\gamma}\|_{\mathcal{A}_K}$. Letting $\gamma \to 0$ the LHS tends to $\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_{\mathcal{A}_K}$, and the RHS tends to $\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_1$. So we get that $\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_{\mathcal{A}_K} = \|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_2}\|_1$, as claimed. ■

3. Proper learning in the well-behaved case

In this section, we focus on properly learning a mixture of k Gaussians under the assumption that we have a "well-behaved" density estimate. We study this case first in order to illustrate our use of shape-restricted polynomials and the $\mathcal{A}_K$-norm. Intuitively, our notion of "well-behavedness" requires that there is a good GMM fit to the density estimate such that the mixture components and the overall mixture distribution live at roughly the same scale. Algorithmically, this allows us to solve our system of polynomial inequalities with sufficient accuracy. In Section 4, we remove this assumption and give another algorithm that works for *all* univariate mixtures of Gaussians and requires no special assumptions on the density estimation algorithm. However, the full algorithm is somewhat more complicated. So for the sake of exposition, the current section is a warm-up and describes the simpler algorithm that works when the density estimate is well-behaved.

3.1. Overview of the Algorithm

The first step of our algorithm is to learn a good piecewise-polynomial approximation p_{dens} for the unknown density f. We achieve this by invoking recent work on density estimation Acharya

et al. (2017). Once we have obtained a good density estimate, it suffices to solve the following optimization problem:

$$\min_{\theta \in \Theta_k} \|p_{\mathrm{dens}} - \mathcal{M}_\theta\|_1 \ .$$

Instead of directly fitting a mixture of Gaussians, we use a mixture of shape-restricted piecewise polynomials as a proxy and solve

$$\min_{\theta \in \Theta_k} \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1 \ .$$

Now all parts of the optimization problem are piecewise polynomials. However, we will see that we cannot directly work with the L_1-norm without increasing the size of the corresponding system of polynomial inequalities substantially. Hence we work with the $\mathcal{A}_K$-norm instead and solve

$$\min_{\theta \in \Theta_k} \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} \ .$$

We approach this problem by converting it to a system of polynomial inequalities with

1. $O(k)$ free variables: one per component weight, mean, and precision,
2. Two levels of quantification: one for the intervals of the $\mathcal{A}_K$-norm, and one for the breakpoints of the shape-restricted polynomial. Each level quantifies over $O(k)$ variables.
3. A Boolean expression on polynomials with $k^{O(k)}$ many constraints.

Finally, we use Renegar's algorithm to approximately solve our system in time $(k \log 1/\epsilon)^{O(k^4)}$. Because we only have to consider the well-behaved case, we know that finding a polynomially good approximation to the parameters will yield a sufficiently close approximation to the true underlying distribution.

3.2. Density estimation, rescaling, and well-behavedness

Density estimation As the first step of our algorithm, we obtain an agnostic estimate of the unknown probability density f. For this, we run the density estimation subroutine ESTIMATE-DENSITY(k, ϵ, δ) from Fact 4. Let p'_{dens} be the resulting $O(k)$-piecewise polynomial. In the following, we condition on the event that

$$\|f - p'_{\mathrm{dens}}\|_1 \ \leq \ 4 \cdot \mathrm{OPT}_k + \epsilon \ .$$

which occurs with probability $1 - \delta$.

Rescaling Since we can solve systems of polynomial inequalities only with bounded precision, we have to post-process the density estimate. For example, it could be the case that some mixture components have extremely large mean parameters μ_i, in which case accurately approximating these parameters could take an arbitrary amount of time. Therefore, we shift and rescale p'_{dens} so that its non-zero part is in $[-1, 1]$ (note that p_{dens} can only have finite support because it consists of a bounded number of pieces).

Let p_{dens} be the scaled and shifted piecewise polynomial. Since the L_1-norm is invariant under shifting and scaling, it suffices to solve the following problem

$$\min_{\theta \in \Theta_k} \|p_{\mathrm{dens}} - \mathcal{M}_\theta\|_1 \ .$$

Once we have solved this problem and found a corresponding θ with

$$\|p_{\text{dens}} - \mathcal{M}_\theta\|_1 \;\leq\; C$$

for some $C \geq 0$, we can undo the transformation applied to the density estimate and get a $\theta' \in \Theta_k$ such that

$$\|p'_{\text{dens}} - \mathcal{M}_{\theta'}\|_1 \;\leq\; C\,.$$

Well-behavedness While rescaling the density estimate p'_{dens} to the interval $[-1, 1]$ controls the size of the mean parameters μ_i, the precision parameters τ_i can still be arbitrarily large. Note that for a mixture component with very large precision, we also have to approximate the corresponding μ_i very accurately. For clarity of presentation, we ignore this issue in this section and assume that the density estimate is *well-behaved*. This assumption allows us to control the accuracy in Renegar's algorithm appropriately. We revisit this point in Section 4 and show how to overcome this limitation. Formally, we introduce the following assumption:

Definition 13 (Well-behaved density estimate) *Let p'_{dens} be a density estimate and let p_{dens} be the rescaled version that is supported on the interval $[-1, 1]$ only. Then we say p_{dens} is γ-well-behaved if there is a set of GMM parameters $\theta \in \Theta_k$ such that*

$$\|p_{\text{dens}} - \mathcal{M}_\theta\|_1 \;=\; \min_{\theta^* \in \Theta_k} \|p_{\text{dens}} - \mathcal{M}_{\theta^*}\|_1$$

and $\tau_i \leq \gamma$ for all $i \in [k]$.

The well-behaved case is interesting in its own right because components with very high precision parameter, i.e., very spiky Gaussians, can often be learnt by clustering the samples.[8] Moreover, the well-behaved case illustrates our use of shape-restricted polynomials and the $\mathcal{A}_K$-distance without additional technical difficulties.

3.3. The $\mathcal{A}_K$-norm as a proxy for the L_1-norm

Computing the L_1-distance between the density estimate p_{dens} and our shape-restricted polynomial approximation $P_{\epsilon,\theta}$ *exactly* requires knowledge of the zeros of the piecewise polynomial $p_{\text{dens}} - P_{\epsilon,\theta}$. In a system of polynomial inequalities, these zeros can be encoded by introducing auxiliary variables. However, note that we cannot simply introduce one variable per zero-crossing without affecting the running time significantly: since the polynomials have degree $O(\log 1/\epsilon)$, this would lead to $O(k \log 1/\epsilon)$ variables, and hence the running time of Renegar's algorithm would depend exponentially on $O(\log 1/\epsilon)$. Such an exponential dependence on $\log(1/\epsilon)$ means that the running time of solving the system of polynomial inequalities becomes super-polynomial in $\frac{1}{\epsilon}$, while our goal was to avoid any polynomial dependence on $\frac{1}{\epsilon}$ when solving the system of polynomial inequalities.

Instead, we use the $\mathcal{A}_K$-norm as an *approximation* of the L_1-norm. Since both $P_{\epsilon,\theta}$ and p_{dens} are close to mixtures of k Gaussians, their difference only has $O(k)$ zero crossings that contribute significantly to the L_1-norm. More formally, we should have $\|p_{\text{dens}} - P_{\epsilon,\theta}\|_1 \approx \|p_{\text{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K}$. And indeed:

8. However, very spiky Gaussians can still be very close, which makes this approach challenging in some cases – see Section 4 for details.

Lemma 14 *Let $\epsilon > 0$, $k \geq 2$, $\theta \in \Theta_k$, and $K = 4k$. Then we have*

$$0 \;\leq\; \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1 - \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} \;\leq\; 8 \cdot \mathrm{OPT}_k + O(\epsilon) \;.$$

Proof Recall Lemma 10: for any function f, we have $\|f\|_{\mathcal{A}_K} \leq \|f\|_1$. Thus, we know that $\|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} \leq \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1$. Hence, it suffices to show that $\|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1 \leq 8 \cdot \mathrm{OPT}_k + O(\epsilon) + \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K}$.

We have conditioned on the event that the density estimation algorithm succeeds. So from Fact 4, we know that there is some mixture of k Gaussians $M_{\theta'}$ so that $\|p_{\mathrm{dens}} - \mathcal{M}_{\theta'}\|_1 \leq 4 \cdot \mathrm{OPT}_k + \epsilon$. By repeated applications of the triangle inequality and Corollary 12, we get

$$\begin{aligned}
\|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1 \;&\leq\; \|p_{\mathrm{dens}} - \mathcal{M}_{\theta'}\|_1 + \|\mathcal{M}_{\theta'} - \mathcal{M}_{\theta}\|_1 + \|P_{\epsilon,\theta} - \mathcal{M}_{\theta}\|_1 \\
&\leq\; 4 \cdot \mathrm{OPT} + \epsilon + \|\mathcal{M}_{\theta'} - \mathcal{M}_{\theta}\|_{\mathcal{A}_K} + \epsilon \\
&\leq\; 4 \cdot \mathrm{OPT} + 2\epsilon + \|\mathcal{M}_{\theta'} - p_{\mathrm{dens}}\|_{\mathcal{A}_K} + \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} + \|P_{\epsilon,\theta} - \mathcal{M}_{\theta}\|_{\mathcal{A}_K} \\
&\leq\; 4 \cdot \mathrm{OPT} + 2\epsilon + \|\mathcal{M}_{\theta'} - p_{\mathrm{dens}}\|_1 + \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} + \|P_{\epsilon,\theta} - \mathcal{M}_{\theta}\|_1 \\
&\leq\; 8 \cdot \mathrm{OPT} + 4\epsilon + \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} \;,
\end{aligned}$$

as claimed. ■

Using this connection between the $\mathcal{A}_K$-norm and the L_1-norm, we can focus our attention on the following problem:

$$\min_{\theta \in \Theta_k} \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} \;.$$

As mentioned above, this problem is simpler from a computational perspective because we only have to introduce $O(k)$ variables into the system of polynomial inequalities, regardless of the value of ϵ.

When encoding the above minimization problem in a system of polynomial inequalities, we convert it to a sequence of feasibility problems. In particular, we solve $O(\log(1/\epsilon))$ feasibility problems of the form

$$\text{Find } \; \theta \in \Theta_k \;\; \text{s.t. } \|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_{\mathcal{A}_K} < \nu \;. \tag{1}$$

Next, we show how to encode such an $\mathcal{A}_K$-constraint in a system of polynomial inequalities.

3.4. A system of polynomial inequalities for encoding closeness in $\mathcal{A}_K$-norm

In this section, we give a general construction for the $\mathcal{A}_K$-distance between any fixed piecewise polynomial (in particular, the density estimate) and any piecewise polynomial we optimize over (in particular, our shape-restricted polynomials which we wish to fit to the density estimate). The only restriction we require is that we already have variables for the breakpoints of the piecewise polynomial we optimize over. As long as these breakpoints depend only polynomially or rationally on the parameters of the shape-restricted piecewise polynomial, this is easy to achieve. Presenting our construction of the $\mathcal{A}_K$-constraints in this generality makes it easy to adapt our techniques to the general algorithm (without the well-behavedness assumption, see Section 4) and to new classes of distributions (see Section 4.5).

The setup in this section will be as follows. Let p be a given, fixed piecewise polynomial supported on $[-1, 1]$ with breakpoints $c_1, \ldots, c_r$. Let $\mathcal{P}$ be a set of piecewise polynomials so that for all $\theta \in S \subseteq \mathbb{R}^u$ for some fixed, known S, there is a $P_\theta(x) \in \mathcal{P}$ with breakpoints $d_1(\theta), \ldots, d_s(\theta)$ such that

- S is a semi-algebraic set.[9] Moreover, assume membership in S can be stated as a Boolean formula over R polynomial predicates, each of degree at most D_1, for some R, D_1.
- For all $1 \leq i \leq s$, there is a polynomial h_i so that $h_i(d_i(\theta), \theta) = 0$, and moreover, for all θ, we have that $d_i(\theta)$ is the unique real number y satisfying $h_i(y, \theta) = 0$. That is, the breakpoints of P_θ can be encoded as polynomial equality in the θ's. Let D_2 be the maximum degree of any h_i.
- The function $(x, \theta) \mapsto P_\theta(x)$ is a polynomial in x and θ as long as x is not at a breakpoint of P_θ. Let D_3 be the maximum degree of this polynomial.

Let $D = \max(D_1, D_2, D_3)$.

Our goal then is to encode the following problem as a system of polynomial inequalities:

$$\text{Find } \theta \in S \text{ s.t. } \|p - P_\theta\|_{\mathcal{A}_K} < \nu \,. \tag{2}$$

In Section 3.5, we show that this is indeed a generalization of the problem in Equation (1), for suitable choices of S and $\mathcal{P}$.

In the following, let $p_\theta^{\text{diff}} \stackrel{\text{def}}{=} p - P_\theta$. Note that p_θ^{diff} is a piecewise polynomial with breakpoints contained in $\{c_1, \ldots c_r, d_1(\theta), \ldots, d_s(\theta)\}$. In order to encode the $\mathcal{A}_K$-constraint, we use the fact that a system of polynomial inequalities can contain for-all quantifiers. Hence it suffices to encode the $\mathcal{A}_K$-constraint for a single set of K intervals. We provide a construction for a single $\mathcal{A}_K$-constraint in Section 3.4.1. In Section 3.4.2, we introduce two further constraints that guarantee validity of the parameters θ and combine these constraints with the $\mathcal{A}_K$-constraint to produce the full system of polynomial inequalities.

3.4.1. ENCODING CLOSENESS FOR A FIXED SET OF INTERVALS

Let $[a_1, b_1], \ldots, [a_K, b_K]$ be K disjoint intervals. In this section we show how to encode the following constraint:

$$\sum_{i=1}^{K} \left| \int_{a_i}^{b_i} p_\theta^{\text{diff}}(x)\, \mathrm{d}x \right| \;\leq\; \nu \,.$$

Note that a given interval $[a_i, b_i]$ might contain several pieces of p_θ^{diff}. In order to encode the integral over $[a_i, b_i]$ correctly, we must therefore know the current order of the breakpoints (which can depend on θ).

However, once the order of the breakpoints of p_θ^{diff} and the a_i and b_i is fixed, the integral over $[a_i, b_i]$ becomes the integral over a fixed set of sub-intervals. Since the integral over a single polynomial piece is still a polynomial, we can then encode this integral over $[a_i, b_i]$ piece-by-piece.

More formally, let Φ be the set of permutations of the variables

$$\{a_1, \ldots, a_K,\, b_1, \ldots, b_K,\, c_1, \ldots, c_r,\, d_1(\theta), \ldots, d_s(\theta)\}$$

9. Recall a semi-algebraic set is a set where membership in the set can be described by polynomial inequalities.

such that (i) the a_i appear in order, (ii) the b_i appear in order, (iii) a_i appears before b_i, and (iv) the c_i appear in order. Let $t = 2K + r + s$. For any $\phi = (\phi_1, \dots, \phi_t) \in \Phi$, let

$$\text{ordered}_{p,\mathcal{P}}(\phi) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{t-1} (\phi_i \le \phi_{i+1}) \ .$$

Note that for any fixed ϕ, this is an unquantified Boolean formula with polynomial constraints in the unknown variables. The order constraints encode whether the current set of variables corresponds to ordered variables under the permutation represented by ϕ. An important property of an ordered ϕ is the following: in each interval $[\phi_i, \phi_{i+1}]$, the piecewise polynomial p_θ^{diff} has exactly one piece. This allows us to integrate over p_θ^{diff} in our system of polynomial inequalities.

Next, we need to encode whether a fixed interval between ϕ_i and ϕ_{i+1} is contained in one of the $\mathcal{A}_K$-intervals, i.e., whether we have to integrate p_θ^{diff} over the interval $[\phi_i, \phi_{i+1}]$ when we compute the $\mathcal{A}_K$-norm of p_θ^{diff}. We use the following expression:

$$\text{is-active}_{p,\mathcal{P}}(\phi, i) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if there is a } j \text{ such that } a_j \text{ appears as or before } \phi_i \text{ in } \phi \\ & \qquad \text{and } b_j \text{ appears as or after } \phi_{i+1} \\ 0 & \text{otherwise} \end{cases} .$$

Note that for fixed ϕ and i, this expression is either 0 or 1 (and hence trivially a polynomial).

With the constructs introduced above, we can now integrate p_θ^{diff} over an interval $[\phi_i, \phi_{i+1}]$. It remains to bound the absolute value of the integral for each individual piece. For this, we introduce a set of t new variables $\xi_1, \dots, \xi_t$ which will correspond to the absolute value of the integral in the corresponding piece.

$$\mathcal{A}_K\text{-bounded-interval}_{p,\mathcal{P}}(\phi, \theta, \xi, i) \stackrel{\text{def}}{=} \left(\left(-\xi_i \le \int_{\phi_i}^{\phi_{i+1}} p_\theta^{\text{diff}}(x)\,\mathrm{d}x \right) \wedge \left(\int_{\phi_i}^{\phi_{i+1}} p_\theta^{\text{diff}}(x)\,\mathrm{d}x \le \xi_i \right) \right) \vee (\text{is-active}_{p,\mathcal{P}}(\phi, i) = 0) \ .$$

Note that the above is a valid polynomial constraint because p_θ^{diff} depends only on θ and x for fixed breakpoint order ϕ and fixed interval $[\phi_i, \phi_{i+1}]$. Moreover, recall that by assumption, $P_{\epsilon,\theta}(x)$ depends polynomially on both θ and x, and therefore the same holds for p_θ^{diff}.

We extend the $\mathcal{A}_K$-check for a single interval to the entire range of p_θ^{diff} as follows:

$$\mathcal{A}_K\text{-bounded-fixed-permutation}_{p,\mathcal{P}}(\phi, \theta, \xi) \stackrel{\text{def}}{=} \bigwedge_{i=1}^{t-1} \mathcal{A}_K\text{-bounded-interval}_{p,\mathcal{P}}(\phi, \theta, \xi, i) \ .$$

We now have all the tools to encode the $\mathcal{A}_K$-constraint for a fixed set of intervals:

$$\mathcal{A}_K\text{-bounded}_{p,\mathcal{P}}(\theta, \nu, a, b, c, d, \xi) \stackrel{\text{def}}{=} \left(\sum_{i=1}^{t-1} \xi_i \le \nu \right) \ \wedge \ \left(\bigwedge_{i=1}^{t-1} (\xi_i \ge 0) \right) \wedge \left(\bigvee_{\phi \in \Phi} \text{ordered}_{p,\mathcal{P}}(\phi) \wedge \mathcal{A}_K\text{-bounded-fixed-permutation}_{p,\mathcal{P}}(\phi, \theta, \xi) \right) \ .$$

By construction, the above constraint now satisfies the following:

Lemma 15 *There exists a vector $\xi \in \mathbb{R}^t$ such that $\mathcal{A}_K$-bounded$_{p,\mathcal{P}}(\theta, \nu, a, b, c, d, \xi)$ is true if and only if*

$$\sum_{i=1}^{K} \left| \int_{a_i}^{b_i} p_\theta^{\text{diff}}(x)\, \mathrm{d}x \right| \;\leq\; \nu \;.$$

Moreover, $\mathcal{A}_K$-bounded$_{p,\mathcal{P}}$ has less than $6t^{t+1}$ polynomial constraints.

The bound on the number of polynomial constraints follows simply from counting the number of polynomial constraints in the construction described above.

3.4.2. COMPLETE SYSTEM OF POLYNOMIAL INEQUALITIES

In addition to the $\mathcal{A}_K$-constraint introduced in the previous subsection, our system of polynomial inequalities contains the following constraints:

Valid parameters First, we encode that the mixture parameters we optimize over are valid, i.e., we let

$$\text{valid-parameters}_S(\theta) \;\stackrel{\text{def}}{=}\; \theta \in S \;.$$

Recall this can be expressed as a Boolean formula over R polynomial predicates of degree at most D.

Correct breakpoints We require that the d_i are indeed the breakpoints of the shape-restricted polynomial P_θ. By the assumption, this can be encoded by the following constraint:

$$\text{correct-breakpoints}_{\mathcal{P}}(\theta, d) \;\stackrel{\text{def}}{=}\; \bigwedge_{i=1}^{s} (h_i(d_i(\theta), \theta) = 0) \;.$$

The full system of polynomial inequalities We now combine the constraints introduced above and introduce our entire system of polynomial inequalities:

$$\begin{aligned} S_{K,p,\mathcal{P},S}(\nu) = \forall a_1, \ldots a_K, b_1, \ldots, b_K \;:& \\ \exists d_1, \ldots, d_s, \xi_1 \ldots \xi_t \;:& \\ \text{valid-parameters}_S(\theta) \;\wedge\; \text{correct-breakpoints}_{\mathcal{P}}(\theta, d) \;\wedge\; \mathcal{A}_K\text{-bounded}_{p,\mathcal{P}}(\theta, \nu, a, b, c, d, \xi) \;.& \end{aligned}$$

This system of polynomial inequalities has

- two levels of quantification, with $2K$ and $s + t$ variables, respectively,
- u free variables,
- $R + s + 4t^{t+1}$ polynomial constraints,
- and maximum degree D in the polynomial constraints.

Let γ be a bound on the free variables, i.e., $\|\theta\|_2 \leq \gamma$, and let λ be a precision parameter. Then Renegar's algorithm (see Fact 5) finds a λ-approximate solution θ for this system of polynomial inequalities satisfying $\|\theta\|_2 \leq \gamma$, if one exists, in time

$$\big((R + s + 6t^{t+1})D\big)^{O(K(s+t)u)} \log\log(3 + \frac{\gamma}{\lambda}) \;.$$

3.5. Instantiating the system of polynomial inequalities for GMMs

We now show how to use the system of polynomial inequalities developed in the previous subsection for our initial goal: that is, encoding closeness between a well-behaved density estimate and a set of shape-restricted polynomials (see Equation 1). Our fixed piecewise polynomial (p in the subsection above) will be p_{dens}. The set of piecewise polynomials we optimize over (the set $\mathcal{P}$ in the previous subsection) will be the set $\mathcal{P}_\epsilon$ of all shape-restricted polynomials $P_{\epsilon,\theta}$. Our S (the domain of θ) will be $\Theta'_k \subseteq \Theta_k$, which we define below. For each $\theta \in S$, we associate it with $P_{\epsilon,\theta}$. Moreover:

- Define

$$\Theta_{k,\gamma} = \left\{ \theta \;\middle|\; \left(\sum_{i=1}^{k} w_i = 1 \right) \wedge \left(\forall\, i \in [k] \;:\; (w_i \geq 0) \wedge (\gamma \geq \tau_i > 0) \wedge (-1 \leq \mu_i \leq 1) \right) \right\} ,$$

  that is, the set of parameters which have bounded means and variances. S is indeed semi-algebraic, and membership in S can be encoded using $2k+1$ polynomial predicates, each with degree $D_1 = 1$.

- For any fixed parameter $\theta \in \Theta_k$, the shape-restricted polynomial P_θ has $s = 2k$ breakpoints by definition, and the breakpoints $d_1(\theta), \ldots, d_{2k}(\theta)$ of $P_{\epsilon,\theta}$ occur at

$$d_{2i-1}(\theta) = \frac{1}{\tau_i}\left(\mu_i - 2\tau_i \log(1/\epsilon)\right) \;, \quad d_{2i}(\theta) = \frac{1}{\tau_i}\left(\mu_i + 2\tau_i \log(1/\epsilon)\right) \;, \text{ for all } 1 \leq i \leq k \;.$$

  Thus, for all parameters θ, the breakpoints $d_1(\theta), \ldots, d_{2k}(\theta)$ are the unique numbers so that so that

$$\tau_i \cdot d_{2i-1}(\theta) - (\mu_i - 2\tau_i \log(1/\epsilon)) = 0 \;, \quad \tau_i \cdot d_{2i}(\theta) - (\mu_i + 2\tau_i \log(1/\epsilon)) = 0 \;, \text{ for all } 1 \leq i \leq k \;,$$

  and thus each of the $d_1(\theta), \ldots, d_{2k}(\theta)$ can be encoded as a polynomial equality of degree $D_2 = 2$.

- Finally, it is straightforward to verify that the map $(x, \theta) \to P_{\epsilon,\theta}(x)$ is a polynomial of degree $D_3 = O(\log 1/\epsilon)$ in (x, θ), at any point where x is not at a breakpoint of P_θ.

From the previous subsection, we know that the system of polynomial inequalities $S_{K,p_{\mathrm{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu)$ has two levels of quantification, each with $O(k)$ variables, it has $k^{O(k)}$ polynomial constraints, and has maximum degree $O(\log 1/\epsilon)$ in the polynomial constraints. Hence, we have shown:

Corollary 16 *For any fixed ϵ, the system of polynomial inequalities $S_{K,p_{\mathrm{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu)$ encodes Equation (1). Moreover, for all $\gamma, \lambda \geq 0$, Renegar's algorithm* SOLVE-POLY-SYSTEM$(S_{K,p_{\mathrm{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu), \lambda, \gamma)$ *runs in time $(k \log(1/\epsilon))^{O(k^4)} \log\log(3 + \frac{\gamma}{\lambda})$.*

3.6. Overall learning algorithm

We now combine our tools developed so far and give an agnostic learning algorithm for the case of well-behaved density estimates (see Algorithm 1).

Algorithm 1 Algorithm for learning a mixture of Gaussians in the well-behaved case.

function LEARN-WELL-BEHAVED-GMM($k, \epsilon, \delta, \gamma$)
  ▷ Density estimation. Only this step draws samples.
  $p'_{\text{dens}} \leftarrow$ ESTIMATE-DENSITY(k, ϵ, δ)
  ▷ Rescaling
  Let p_{dens} be a rescaled and shifted version of p'_{dens} such that the support of p_{dens} is $[-1, 1]$.
  Let α and β be such that $p_{\text{dens}}(x) = p'_{\text{dens}}\left(\frac{2(x-\alpha)}{\beta-\alpha} - 1\right)$
  ▷ Fitting shape-restricted polynomials
  $K \leftarrow 4k$
  $\nu \leftarrow \epsilon$
  $\theta \leftarrow$ SOLVE-POLY-SYSTEM$(S_{K,p_{\text{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu), C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2, 3k\gamma)$
  while θ is "NO-SOLUTION" **do**
    $\nu \leftarrow 2 \cdot \nu$
    $\theta \leftarrow$ SOLVE-POLY-SYSTEM$(S_{K,p_{\text{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu), C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2, 3k\gamma)$
  ▷ Fix the parameters
  for $i = 1, \ldots, k$ **do**
    if $\tau_i \le 0$, set $w_i \leftarrow 0$ and set τ_i to be arbitrary but positive.
  Let $W = \sum_{i=1}^k w_i$
  for $i = 1, \ldots, k$ **do**
    $w_i \leftarrow w_i / W$
  ▷ Undo the scaling
  $w'_i \leftarrow w_i$
  $\mu'_i \leftarrow \frac{(\mu_i+1)(\beta-\alpha)}{2} + \alpha$
  $\tau'_i \leftarrow \frac{\tau_i}{\beta-\alpha}$
  return θ'

3.7. Analysis

Before we prove correctness of LEARN-WELL-BEHAVED-GMM, we introduce two auxiliary lemmas.

An important consequence of the well-behavedness assumption (see Definition 13) are the following robustness properties.

Lemma 17 (Parameter stability) *Fix $2 \ge \epsilon > 0$. Let the parameters $\theta, \theta' \in \Theta_k$ be such that (i) $\tau_i, \tau'_i \le \gamma$ for all $i \in [k]$ and (ii) $\|\theta - \theta'\|_2 \le C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2$, for some universal constant C. Then*

$$\|\mathcal{M}_\theta - \mathcal{M}_{\theta'}\|_1 \;\le\; \epsilon\,.$$

Before we prove this lemma, we first need a calculation which quantifies the robustness of the standard normal pdf to small perturbations.

Lemma 18 *For all $2 \ge \epsilon > 0$, there is a $\delta_1 = \delta_1(\epsilon) = \frac{\epsilon}{20\sqrt{\log(1/\epsilon)}} \ge O(\epsilon^2)$ so that for all $\delta \le \delta_1$, we have $\|\mathcal{N}(x) - \mathcal{N}(x+\delta)\|_1 \le O(\epsilon)$.*

Proof Note that if $\epsilon > 2$ this claim holds trivially for all choices of δ since the L_1-distance between two distributions can only ever be 2. Thus assume that $\epsilon \leq 2$. Let I be an interval centered at 0 so that both $\mathcal{N}(x)$ and $\mathcal{N}(x+\delta)$ assign $1 - \frac{\epsilon}{2}$ weight on this interval. By standard properties of Gaussians, we know that $|I| \leq 10\sqrt{\log(1/\epsilon)}$. We thus have

$$\|\mathcal{N}(x) - \mathcal{N}(x+\delta)\|_1 \leq \int_I |\mathcal{N}(x) - \mathcal{N}(x+\delta)|\, dx + \epsilon \; .$$

By Taylor's theorem we have that for all x,

$$\left| e^{-(x+\delta)^2/2} - e^{-x^2/2} \right| \leq C \cdot \delta$$

for some universal constant $C = \max_{x \in \mathbb{R}} \frac{\mathrm{d}}{\mathrm{d}x}(e^{-\frac{x^2}{2}}) \leq 1$. Since we choose $\delta_1 \leq \frac{\epsilon}{20\sqrt{\log(1/\epsilon)}}$, we must have that

$$\|\mathcal{N}(x) - \mathcal{N}(x+\delta)\|_1 \leq O(\epsilon) \; ,$$

as claimed. ■

Proof [Proof of Lemma 17] Notice the ℓ_2 guarantee of Renegar's algorithm (see Fact 5) also trivially implies an ℓ_∞ guarantee on the error in the parameters θ; that is, for all i, we will have that the weights, means, and variances of the two components differ by at most $C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2$. By repeated applications of the triangle inequality to the quantity in the lemma, it suffices to show the three following claims:

- For any μ, τ,
$$\|w_1 \mathcal{N}_{\mu,\tau}(x) - w_2 \mathcal{N}_{\mu,\tau}(x)\|_1 \leq \frac{\epsilon}{3k}$$
if $|w_1 - w_2| \leq C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2$.
- For any $\tau \leq \gamma$,
$$\|\mathcal{N}_{\mu_1,\tau}(x) - \mathcal{N}_{\mu_2,\tau}(x)\|_1 \leq \frac{\epsilon}{3k}$$
if $|\mu_1 - \mu_2| \leq C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2$.
- For any μ,
$$\|\mathcal{N}_{\mu,\tau_1}(x) - \mathcal{N}_{\mu,\tau_2}(x)\|_1 \leq \frac{\epsilon}{3k}$$
if $|\tau_1 - \tau_2| \leq C\frac{1}{\gamma}\left(\frac{\epsilon}{k}\right)^2$.

The first inequality is trivial, for C sufficiently small. The second and third inequalities follow from a change of variables and an application of Lemma 18. ■

Recall that our system of polynomial inequalities only considers mean parameters in $[-1, 1]$. The following lemma shows that this restriction still allows us to find a good approximation once the density estimate is rescaled to $[-1, 1]$.

Lemma 19 (Restricted means) *Let $g : \mathbb{R} \to \mathbb{R}$ be a function supported on $[-1, 1]$, i.e., $g(x) = 0$ for $x \notin [-1, 1]$. Moreover, let $\theta^* \in \Theta_k$. Then there is a $\theta' \in \Theta_k$ such that $\mu'_i \in [-1, 1]$ for all $i \in [k]$ and*

$$\|g - \mathcal{M}_{\theta'}\|_1 \;\leq\; 5 \cdot \|g - \mathcal{M}_{\theta^*}\|_1 \,.$$

Proof Let $A = \{i \,|\, \mu^*_i \in [-1, 1]\}$ and $B = [k] \setminus A$. Let θ' be defined as follows:

- $w'_i = w^*_i$ for all $i \in [k]$.
- $\mu'_i = \mu^*_i$ for $i \in A$ and $\mu'_i = 0$ for $i \in B$.
- $\tau'_i = \tau^*_i$ for all $i \in [k]$.

From the triangle inequality, we have

$$\|g - \mathcal{M}_{\theta'}\|_1 \;\leq\; \|g - \mathcal{M}_{\theta^*}\|_1 + \|\mathcal{M}_{\theta^*} - \mathcal{M}_{\theta'}\|_1 \,. \tag{3}$$

Hence it suffices to bound $\|\mathcal{M}_{\theta^*} - \mathcal{M}_{\theta'}\|_1$.

Note that for $i \in B$, the corresponding i-th component has at least half of its probability mass outside $[-1, 1]$. Since g is zero outside $[-1, 1]$, this mass of the i-th component must therefore contribute to the error $\|g - \mathcal{M}_{\theta^*}\|_1$. Let $\mathbb{1}[x \notin [-1, 1]]$ be the indicator function of the set $\mathbb{R} \setminus [-1, 1]$. Then we get

$$\|g - \mathcal{M}_{\theta^*}\|_1 \;\geq\; \|\mathcal{M}_{\theta^*} \cdot \mathbb{1}[x \notin [-1, 1]]\|_1 \;\geq\; \frac{1}{2}\left\|\sum_{i \in B} w^*_i \cdot \mathcal{N}_{\mu^*_i, \tau^*_i}\right\|_1 \,.$$

For $i \in A$, the mixture components of $\mathcal{M}_{\theta^*}$ and $\mathcal{M}_{\theta'}$ match. Hence we have

$$\begin{aligned}
\|\mathcal{M}_{\theta^*} - \mathcal{M}_{\theta'}\|_1 \;&=\; \left\|\sum_{i \in B} w^*_i \cdot \mathcal{N}_{\mu^*_i, \tau^*_i} \;-\; \sum_{i \in B} w'_i \cdot \mathcal{N}_{\mu'_i, \tau'_i}\right\|_1 \\
&\leq\; \left\|\sum_{i \in B} w^*_i \cdot \mathcal{N}_{\mu^*_i, \tau^*_i}\right\|_1 \;+\; \left\|\sum_{i \in B} w'_i \cdot \mathcal{N}_{\mu'_i, \tau'_i}\right\|_1 \\
&=\; 2 \cdot \left\|\sum_{i \in B} w^*_i \cdot \mathcal{N}_{\mu^*_i, \tau^*_i}\right\|_1 \\
&\leq\; 4 \cdot \|g - \mathcal{M}_{\theta^*}\|_1 \,.
\end{aligned}$$

Combining this inequality with (3) gives the desired result. ■

We now prove our main theorem for the well-behaved case.

Theorem 20 *Let $\delta, \epsilon, \gamma > 0$, $k \geq 1$, and let f be the pdf of the unknown distribution. Moreover, assume that the density estimate p'_{dens} obtained in Line 3 of Algorithm 1 is γ-well-behaved. Then the algorithm* LEARN-WELL-BEHAVED-GMM$(k, \epsilon, \delta, \gamma)$ *returns a set of GMM parameters θ' such that*

$$\|\mathcal{M}_{\theta'} - f\|_1 \;\leq\; 60 \cdot \mathrm{OPT}_k + \epsilon$$

with probability $1 - \delta$. Moreover, the algorithm runs in time

$$\left(k \cdot \log \frac{1}{\epsilon}\right)^{O(k^4)} \cdot \log \frac{1}{\epsilon} \cdot \log\log \frac{k\gamma}{\epsilon} \;+\; \widetilde{O}\left(\frac{k}{\epsilon^2}\right) \,.$$

Proof First, we prove the claimed running time. From Fact 4, we know that the density estimation step has a time complexity of $\widetilde{O}(\frac{k}{\epsilon^2})$. Next, consider the second stage where we fit shape-restricted polynomials to the density estimate. Note that for $\nu = 3$, the system of polynomial inequalities $S_{p_{\mathrm{dens}},\mathcal{P}_\epsilon}(\nu)$ is trivially satisfiable because the $\mathcal{A}_K$-norm is bounded by the L_1-norm and the L_1-norm between the two (approximate) densities is at most $2 + O(\epsilon)$. Hence the while-loop in the algorithm takes at most $O(\log \frac{1}{\epsilon})$ iterations. Combining this bound with the size of the system of polynomial inequalities (see Subsection 3.4.2) and the time complexity of Renegar's algorithm (see Fact 5), we get the following running time for solving all systems of polynomial inequalities proposed by our algorithm:

$$\left(k \cdot \log \frac{1}{\epsilon}\right)^{O(k^4)} \cdot \log\log \frac{k\gamma}{\epsilon} \cdot \log \frac{1}{\epsilon} \,.$$

This proves the stated running time.

Next, we consider the correctness guarantee. We condition on the event that the density estimation stage succeeds, which occurs with probability $1 - \delta$ (Fact 4). Then we have

$$\|f - p'_{\mathrm{dens}}\|_1 \le 4 \cdot \mathrm{OPT}_k + \epsilon \,.$$

By assumption, the rescaled density estimate p_{dens} is γ-well-behaved. Recalling Definition 13, this means that there is a set of GMM parameters $\theta \in \Theta_k$ such that $\tau_i \le \gamma$ for all $i \in [k]$ and

$$\begin{aligned}
\|p_{\mathrm{dens}} - \mathcal{M}_\theta\|_1 &= \min_{\theta^* \in \Theta_k} \|p_{\mathrm{dens}} - \mathcal{M}_{\theta^*}\|_1 \\
&= \min_{\theta^* \in \Theta_k} \|p'_{\mathrm{dens}} - \mathcal{M}_{\theta^*}\|_1 \\
&\le \min_{\theta^* \in \Theta_k} \|p'_{\mathrm{dens}} - f\|_1 + \|f - \mathcal{M}_{\theta^*}\|_1 \\
&\le 4 \cdot \mathrm{OPT}_k + \epsilon + \min_{\theta^* \in \Theta_k} \|f - \mathcal{M}_{\theta^*}\|_1 \\
&\le 5 \cdot \mathrm{OPT}_k + \epsilon \,.
\end{aligned}$$

Applying the triangle inequality again, this implies that

$$\|p_{\mathrm{dens}} - P_{\epsilon,\theta}\|_1 \le \|p_{\mathrm{dens}} - \mathcal{M}_\theta\|_1 + \|\mathcal{M}_\theta - P_{\epsilon,\theta}\|_1 \le 5 \cdot \mathrm{OPT}_k + 2\epsilon \,.$$

This almost implies that $S_{p_{\mathrm{dens}},\mathcal{P}_\epsilon}(\nu)$ is feasible for $\nu \ge 5 \cdot \mathrm{OPT}_k + 2\epsilon$. However, there are two remaining steps. First, recall that the system of polynomial inequalities restricts the means to lie in $[-1, 1]$. Hence we use Lemma 19, which implies that there is a $\tilde{\theta} \in \Theta_k$ such that $\tilde{\mu}_i \in [-1, 1]$ and

$$\|p_{\mathrm{dens}} - P_{\epsilon,\tilde{\theta}}\|_1 \le 25 \cdot \mathrm{OPT}_k + 10\epsilon \,.$$

Moreover, the system of polynomial inequalities works with the $\mathcal{A}_K$-norm instead of the L_1-norm. Using Lemma 14, we get that

$$\|p_{\mathrm{dens}} - P_{\epsilon,\tilde{\theta}}\|_{\mathcal{A}_K} \le \|p_{\mathrm{dens}} - P_{\epsilon,\tilde{\theta}}\|_1 \,.$$

Therefore, in some iteration when

$$\nu \le 2 \cdot (25 \cdot \mathrm{OPT}_k + 10\epsilon) = 50 \cdot \mathrm{OPT}_k + 20\epsilon$$

the system of polynomial inequalities $S_{p_{\text{dens}},\mathcal{P}_\epsilon,\Theta_{k,\gamma}}(\nu)$ become feasible and Renegar's algorithm guarantees that we find parameters θ' such that $\|\theta' - \theta^\dagger\|_2 \leq \frac{\epsilon}{\gamma}$ for some $\theta^\dagger \in \Theta_k$ and

$$\|p_{\text{dens}} - \mathcal{M}_{\theta^\dagger}\|_{\mathcal{A}_K} \;\leq\; 50 \cdot \text{OPT}_k + O(\epsilon) \;.$$

Note that we used well-behavedness here to ensure that the precisions in $\theta^\dagger$ are bounded by γ. Let θ be the parameters we return. It is not difficult to see that $\|\theta - \theta^\dagger\|_2 \leq \frac{2\epsilon}{\gamma}$. We convert this back to an L_1 guarantee via Lemma 14:

$$\|p_{\text{dens}} - \mathcal{M}_{\theta^\dagger}\|_1 \;\leq\; 56 \cdot \text{OPT}_k + O(\epsilon) \;.$$

Next, we use parameter stability (Lemma 17) and get

$$\|p_{\text{dens}} - \mathcal{M}_{\theta}\|_1 \;\leq\; 56 \cdot \text{OPT}_k + O(\epsilon) \;.$$

We now relate this back to the unknown density f. Let θ' be the parameters θ scaled back to the original density estimate (see Lines 21 to 23 in Algorithm 1). Then we have

$$\|p'_{\text{dens}} - \mathcal{M}_{\theta'}\|_1 \;\leq\; 56 \cdot \text{OPT}_k + O(\epsilon) \;.$$

Using the fact that p'_{dens} is a good density estimate, we get

$$\begin{aligned}\|f - \mathcal{M}_{\theta'}\|_1 \;&\leq\; \|f - p'_{\text{dens}}\|_1 \;+\; \|p'_{\text{dens}} - \mathcal{M}_{\theta'}\|_1 \\ &\leq\; 4 \cdot \text{OPT}_k + \epsilon \;+\; 56 \cdot \text{OPT}_k + O(\epsilon) \\ &\leq\; 60 \cdot \text{OPT}_k + O(\epsilon) \;.\end{aligned}$$

As a final step, we choose an internal ϵ' in our algorithm so that the $O(\epsilon')$ in the above guarantee becomes bounded by ϵ. This proves the desired approximation guarantee. ■

4. General algorithm for the univariate case

4.1. Preliminaries

As before, we let p_{dens} be the piecewise polynomial returned by LEARN-PIECEWISE-POLYNOMIAL (see Fact 4). Let $I_0, \ldots, I_{s+1}$ be the intervals defined by the breakpoints of p_{dens}. Recall that p_{dens} has degree $O(\log 1/\epsilon)$ and has $s + 2 = O(k)$ pieces. Furthermore, I_0 and I_{s+1} are unbounded in length, and on these intervals p_{dens} is zero. By rescaling and translating, we may assume WLOG that $\cup_{i=1}^{s} I_i$ is $[-1, 1]$.

Recall that $\mathcal{I}$ is defined by the set of intervals $\{I_1, \ldots, I_s\}$. We know that $s = O(k)$. Intuitively, these intervals capture the different scales at which we need to operate. We formalize this intuition below.

Definition 21 *For any Gaussian $\mathcal{N}_{\mu,\tau}$, let $L(\mathcal{N}_{\mu,\tau})$ be the interval centered at μ on which $\mathcal{N}_{\mu,\tau}$ places exactly W of its weight, where $0 < W < 1$ is a universal constant we will determine later. By properties of Gaussians, there is some absolute constant $\omega > 0$ such that $\mathcal{N}_{\mu,\tau}(x) \geq \omega\tau$ for all $x \in L(\mathcal{N}_{\mu,\tau})$.*

Definition 22 *Say a Gaussian $\mathcal{N}_{\mu,\tau}$ is* admissible *if (i) $\mathcal{N}_{\mu,\tau}$ places at least $1/2$ of its mass in $[-1,1]$, and (ii) there is a $J \in \mathcal{I}$ so that $|J \cap L(\mathcal{N}_{\mu,\tau})| \geq 1/(8s\tau)$ and so that*

$$\tau \leq \frac{1}{|J|} \cdot \phi,$$

where

$$\phi \;=\; \phi(\epsilon, k) \;\stackrel{def}{=}\; \frac{32k}{\omega\epsilon} m(m+1)^2 \cdot (\sqrt{2}+1)^m \;,$$

where m is the degree of p_{dens}. We call the interval $J \in \mathcal{I}$ satisfying this property on which $\mathcal{N}_{\mu,\tau}$ places most of its mass its associated interval.

Fix $\theta \in \Theta_k$. We say the ℓ-th component is admissible *if the underlying Gaussian is admissible and moreover $w_\ell \geq \epsilon/k$.*

Notice that since $m = O(\log(1/\epsilon))$, we have that $\phi(\epsilon, k) = \mathrm{poly}(1/\epsilon, k)$.

Lemma 23 (No Interaction Lemma) *Fix $\theta \in \Theta_k$. Let $S_{\mathrm{good}}(\theta) \subseteq [k]$ be the set of $\ell \in [k]$ whose corresponding mixture component is admissible, and let $S_{\mathrm{bad}}(\theta)$ be the rest. Then, we have*

$$\|\mathcal{M}_\theta - p_{\mathrm{dens}}\|_1 \geq \left\| \sum_{\ell \in S_{\mathrm{good}}(\theta)} w_\ell \cdot \mathcal{N}_{\mu_\ell,\tau_\ell} - p_{\mathrm{dens}} \right\|_1 + \frac{1}{2} \sum_{\ell \in S_{\mathrm{bad}}(\theta)} w_\ell \; - 2\epsilon.$$

We briefly remark that the constant $\frac{1}{2}$ we obtain here is somewhat arbitrary; by choosing different universal constants above, one can obtain any fraction arbitrarily close to one, at a minimal loss.

Proof Fix $\ell \in S_{\mathrm{bad}}(\theta)$, and denote the corresponding component $\mathcal{N}_\ell$. Recall that it has mean μ_ℓ and precision τ_ℓ. Let $L_\ell = L(\mathcal{N}_\ell)$.

Let $\mathcal{M}_\theta^{-\ell}(x) = \sum_{i \neq \ell} w_i \mathcal{N}_{\mu_i,\tau_i}(x)$ be the density of the mixture without the ℓ-th component. We will show that

$$\|\mathcal{M}_\theta - p_{\mathrm{dens}}\|_1 \geq \|M_\theta^{-\ell} - p_{\mathrm{dens}}\|_1 + \frac{1}{2} w_\ell - \frac{2\epsilon}{k}.$$

It suffices to prove this inequality because then we may repeat the argument with a different $\ell' \in S_{\mathrm{bad}}(\theta)$ until we have subtracted out all such ℓ, and this yields the claim in the lemma.

If $w_\ell \leq \epsilon/k$ then this statement is obvious. If $\mathcal{N}_\ell$ places less than half its weight on $[-1,1]$, then this is also obvious. Thus we will assume that $w_\ell > \epsilon/k$ and $\mathcal{N}_\ell$ places at least half its weight on $[-1,1]$.

Let $\mathcal{I}_\ell$ be the set of intervals in $\mathcal{I}$ which intersect L_ℓ. We partition the intervals in $\mathcal{I}_\ell$ into two groups:

1. Let $\mathcal{L}_1$ be the set of intervals $J \in \mathcal{I}_\ell$ so that $|J \cap L_\ell| \leq 1/(8s\tau_\ell)$.

2. Let $\mathcal{L}_2$ be the set of intervals $J \in \mathcal{I}_\ell$ not in $\mathcal{L}_1$ so that

$$\tau_\ell > \frac{1}{|J|} \cdot \phi \;.$$

By the definition of admissibility, this is indeed a partition of $\mathcal{I}_\ell$.

We have

$$\begin{aligned}
\|\mathcal{M}_\theta - p_{\text{dens}}\|_1 &= \left\|\mathcal{M}_\theta^{-\ell} + \mathcal{N}_\ell - p_{\text{dens}}\right\|_1 \\
&= \int_{L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x + \int_{L_\ell^c} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x \\
&\geq \int_{L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x + \int_{L_\ell^c} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x - w_\ell \int_{L_\ell^c} \mathcal{N}_\ell(x)\,\mathrm{d}x \\
&\geq \int_{L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x + \int_{L_\ell^c} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x - (1-W) w_\ell \,.
\end{aligned}$$

We split the first term on the RHS into two parts, given by our partition:

$$\begin{aligned}
\int_{L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x &= \sum_{J \in \mathcal{L}_1} \int_{J \cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x \\
&\quad + \sum_{J \in \mathcal{L}_2} \int_{J \cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x \,.
\end{aligned}$$

We lower bound the contribution of each term separately.

(1) We first bound the first term. Since for each $J \in \mathcal{L}_1$ we have $|J \cap L_\ell| \leq 1/(8s\tau_\ell)$, we know that

$$\int_{J \cap L_\ell} \mathcal{N}_\ell(x)\,\mathrm{d}x \leq \frac{1}{8s} \tag{4}$$

and so

$$\begin{aligned}
\sum_{J \in \mathcal{L}_1} \int_{J \cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x &\geq \sum_{J \in \mathcal{L}_1} \int_{J \cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x - |\mathcal{L}_1| \cdot w_\ell \cdot \frac{1}{8s} \\
&\geq \sum_{J \in \mathcal{L}_1} \int_{J \cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x - \frac{1}{8} w_\ell
\end{aligned}$$

since $\mathcal{I}$ and thus $\mathcal{L}_1$ contains at most s intervals.

(2) We now consider the second term. Fix a $J \in \mathcal{L}_2$, and let p_J be the polynomial which is equal to p_{dens} on J. Since $\int p_{\text{dens}} \leq 1 + \epsilon \leq 2$ (as otherwise its L_1-distance to the unknown density would be more than ϵ) and p_{dens} is nonnegative, we also know that $\int_J p_J \leq 2$. We require the following fact (see Acharya et al. (2017)):

Fact 24 *Let $p(x) = \sum_{j=0}^m c_j x^j$ be a degree-m polynomial so that $p \geq 0$ on $[-1,1]$ and $\int_{-1}^1 p \leq \beta$. Then $\max_i |c_i| \leq \beta \cdot (m+1)^2 \cdot (\sqrt{2}+1)^m$.*

Consider the shifted polynomial $q_J(u) = p_J(u \cdot (b_J - a_J)/2 + (b_J + a_J)/2)$ where $J = [a_J, b_J]$. By applying Fact 24 to q_J and noting that $\int_{-1}^1 q_J = (2/|J|) \cdot \int_J p_J$, we conclude that the coefficients of q_J are bounded by

$$\frac{4}{|J|} \cdot (m+1)^2 \cdot (\sqrt{2}+1)^m$$

and thus

$$|q_J(u)| \le \frac{4}{|J|} \cdot m(m+1)^2 \cdot (\sqrt{2}+1)^m$$

for all $u \in [-1,1]$, and so therefore the same bound applies for $p_J(x)$ for all $x \in J$.

But notice that since we assume that $J \in \mathcal{L}_2$, it follows that for all $x \in J \cap L_\ell$, we have that

$$\mathcal{N}_\ell(x) \ge 8\frac{k}{\epsilon} p_J(x) \ ,$$

and so in particular $w_\ell \mathcal{N}_\ell(x) \ge 8p_J(x)$ for all $x \in J \cap L_\ell$. Hence we have

$$\begin{aligned}
\int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x &= \int_{J\cap L_\ell} \mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_J(x)\,\mathrm{d}x \\
&\ge \int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_J(x)\right| \mathrm{d}x + \int_{J\cap L_\ell} \frac{7}{8} w_\ell \mathcal{N}_\ell(x) - p_J(x)\,\mathrm{d}x \\
&\ge \int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_J(x)\right| \mathrm{d}x + \frac{3w_\ell}{4} \int_{J\cap L_\ell} \mathcal{N}_\ell(x)\,\mathrm{d}x \ .
\end{aligned}$$

where the second line follows since $\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell - p_J(x) \ge \left|\mathcal{M}_\theta^{-\ell}(x) - p_J(x)\right| + \frac{7}{8} w_\ell \mathcal{N}_\ell(x) - p_J(x)$ for all $x \in J \cap L_\ell$.

Thus

$$\begin{aligned}
\sum_{J\in\mathcal{L}_2} \int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) + w_\ell \mathcal{N}_\ell(x) - p_{\text{dens}}(x)\right| \mathrm{d}x \ \ge \\
\sum_{J\in\mathcal{L}_2} \left(\int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x + \frac{3w_\ell}{4} \int_{J\cap L_\ell} \mathcal{N}_\ell(x)\,\mathrm{d}x \right) \ .
\end{aligned} \tag{5}$$

Moreover, by Equation (4), we know that

$$\begin{aligned}
\sum_{J\in\mathcal{L}_2} \int_{J\cap L_\ell} \mathcal{N}_\ell(x)\,\mathrm{d}x &= \int_{L_\ell} \mathcal{N}_\ell(x)dx - \sum_{J\in\mathcal{L}_1} \int_{J\cap L_\ell} \mathcal{N}_\ell(x)\,\mathrm{d}x \\
&\ge W - \frac{1}{8} \ ,
\end{aligned}$$

since $\mathcal{L}_1$ contains at most s intervals. Thus, the RHS of Equation (5) must be lower bounded by

$$\sum_{J\in\mathcal{L}_2} \int_{J\cap L_\ell} \left|\mathcal{M}_\theta^{-\ell}(x) - p_{\text{dens}}(x)\right| \mathrm{d}x + \frac{3}{4}\left(W - \frac{1}{8}\right) w_\ell \ .$$

Putting it all together. Hence, we have

$$\begin{aligned}\int_{L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x &= \sum_{J\in\mathcal{L}_1}\int_{J\cap L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \sum_{J\in\mathcal{L}_2}\int_{J\cap L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x \\ &\geq \sum_{J\in\mathcal{L}_1}\int_{J\cap L_\ell} |M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \sum_{J\in\mathcal{L}_2}\int_{J\cap L_\ell} |M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x \\ &\qquad + \left[\frac{3}{4}\left(W - \frac{1}{8}\right) - \frac{1}{8}\right] w_\ell \\ &\geq \int_{L_\ell} |M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \left[\frac{3}{4}\left(W - \frac{1}{8}\right) - \frac{1}{8}\right] w_\ell\,.\end{aligned}$$

We therefore have

$$\begin{aligned}\|M_\theta - p_{\mathrm{dens}}\|_1 &= \int_{L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \int_{L_\ell^c} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x \\ &\geq \int_{L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \int_{L_\ell^c} \left|M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)\right|\mathrm{d}x - \int_{L_\ell^c} w_\ell \mathcal{N}_\ell(x)\,\mathrm{d}x \\ &\geq \int_{L_\ell} |M_\theta(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \int_{L_\ell^c} \left|M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)\right|\mathrm{d}x - (1-W)w_\ell \\ &\geq \int_{L_\ell} |M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)|\,\mathrm{d}x + \left(\frac{7}{4}W - \frac{39}{32}\right) w_\ell + \int_{L_\ell^c} \left|M_\theta^{-\ell}(x) - p_{\mathrm{dens}}(x)\right|\mathrm{d}x \\ &= \|M_\theta^{-\ell} - p_{\mathrm{dens}}\|_1 + \frac{1}{2}w_\ell\,,\end{aligned}$$

when we set $W = 55/56$. ■

4.2. A parametrization scheme for a single Gaussian

Intuitively, Lemma 23 says that for any $\theta \in \Theta_k$, there are some components which have bounded variance and which can be close to p_{dens} (the components in $S_{\mathrm{good}}(\theta)$), and the remaining components, which may have unbounded variance but which will be far away from p_{dens}. Since we are searching for a k-GMM which is close to p_{dens}, in some sense we should not have to concern ourselves with the latter components since they cannot meaningfully interact with p_{dens}. Thus we only need find a suitably robust parametrization for admissible Gaussians.

Such a parametrization can be obtained by linearly transforming the domain so that the associated interval gets mapped to $[-1, 1]$. Formally, fix a Gaussian $\mathcal{N}_{\mu,\tau}$ and an interval J. Then it can be written as

$$\mathcal{N}_{\mu,\tau}(x) = \frac{\tilde{\tau}}{|J|/2}\mathcal{N}\left(\tilde{\tau}\cdot\frac{x - \mathrm{mid}(J)}{|J|/2} - \tilde{\mu}\right)\,, \tag{6}$$

for some unique $\tilde{\mu}$ and $\tilde{\tau}$, where for any interval I, we define $\mathrm{mid}(I)$ to denote its midpoint. Call these the *rescaled mean with respect to* J and *rescaled precision with respect to* J of $\mathcal{N}$, respectively. Concretely, given μ, τ, and an interval J, the rescaled variance and mean with respect to J are defined to be

$$\tilde{\tau} = \frac{|J|}{2}\tau\,, \qquad \tilde{\mu} = \frac{\tilde{\tau}}{|J|/2}\left(\mu - \mathrm{mid}(J)\right)\,.$$

For any $\tilde{\mu}, \tilde{\tau}$, we let $\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}(x)$ denote the function given by the RHS of Equation (6). The following two lemmas says that these rescaled parameters have the desired robustness properties.

Lemma 25 *Let $\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}$ be an admissible Gaussian with rescaled mean $\tilde{\mu}$ and rescaled precision $\tilde{\tau}$ with respect to its associated interval $J \in \mathcal{I}$. Then $\tilde{\mu} \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$ and $\sqrt{2\pi}\cdot\omega/(16s) \leq \tilde{\tau} \leq \phi/2$.*

Proof We first show that $\sqrt{2\pi}\cdot\omega/(16s) \leq \tilde{\tau} \leq \phi/2$. That the rescaled variance is bounded from above follows from a simple change of variables and the definition of admissibility. By the definition of admissibility, we also know that

$$\begin{aligned}\int_J \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x &\geq \int_{J\cap L(\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}})} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \\ &\geq \omega\tau\cdot|J\cap L(\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}})| \\ &\geq \frac{\omega}{8s}\,.\end{aligned}$$

Furthermore, we trivially have

$$|J|\cdot\frac{\tau}{\sqrt{2\pi}} \;\geq\; \int_J \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x\,.$$

Thus, the precision τ must be at least $\sqrt{2\pi}\omega/(8s|J|)$, and so its rescaled precision must be at least $\sqrt{2\pi}\omega/(16s)$, as claimed.

We now show that $\tilde{\mu} \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$. Because $\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}$ is an admissible Gaussian with associated interval J, we know that $|J\cap L(\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}})| \geq 1/(8s\tau)$. Moreover, we know that on $J\cap L(\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}})$, we have $\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}(x) \geq \omega\tau$. Thus in particular

$$\int_J \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \geq \int_{J\cap L(\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}})} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \geq \frac{\omega}{8s}\,.$$

Define $\tilde{J}$ to be the interval which is of length $8s|J|/\omega$ around $\mathrm{mid}(J)$. We claim that $\mu \in \tilde{J}$, where μ is the mean of $\mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}$.

Assume that $\mathrm{mid}(J) \leq \mu$. Let $J_0 = J$ and inductively, for $i < 4s/\omega$, let J_i be the interval with left endpoint at the right endpoint of J_{i-1} and with length $|J|$. That is, the J_i consist of $4s/\omega$ consecutive, non-intersecting copies of J starting at J and going upwards on the number line (for simplicity of exposition we assume that $4s/\omega$ is an integer). Let $J^\dagger = \cup_{i=0}^{(4s/\omega)-1} J_i$. We claim that $\mu \in J^\dagger$. Suppose not. This means that μ is strictly greater than any point in any J_i. In particular, this implies that for all i,

$$\begin{aligned}\int_{J_i} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x &\geq \int_{J_{i-1}} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \\ &\geq \int_{J_0} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \\ &\geq \frac{\omega}{8s}\,.\end{aligned}$$

But then this would imply that

$$\int_{J^\dagger} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x = \sum_{i=0}^{(4s/\omega)-1} \int_{J_i} \mathcal{N}^{r,J}_{\tilde{\mu},\tilde{\tau}}\,\mathrm{d}x \geq \frac{1}{2}\,.$$

Notice that $J^\dagger$ is itself an interval. But any interval containing at least $1/2$ of the weight of any Gaussian must contain its mean, which we assumed did not happen. Thus we conclude that $\mu \in J^\dagger$. Moreover, $J^\dagger \subseteq \tilde{J}$, so $\mu \in \tilde{J}$, as claimed. If $\mathrm{mid}(J) \geq \mu$ then apply the symmetric argument with J_i which are decreasing on the number line instead of increasing.

We have thus shown that $\mu \in \tilde{J}$. It is a straightforward calculation to show that this implies that $\tilde{\mu} \in [-\frac{4s\tau}{\omega}, \frac{4s\tau}{\omega}]$. By the above, we know that $\tau \leq \phi/2$ and thus $\tilde{\mu} \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$, as claimed. ■

Lemma 26 *For any interval J, and $\tilde{\mu}_1, \tilde{\tau}_1, \tilde{\mu}_2, \tilde{\tau}_2$ so that $|\tilde{\tau}_i| \leq 2\phi$ for $i \in \{1,2\}$ and $|\tilde{\mu_1} - \tilde{\mu}_2| + |\tilde{\tau}_1 - \tilde{\tau}_2| \leq O((\epsilon/(\phi k))^2)$, we have*

$$\|\mathcal{N}^{r,J}_{\tilde{\mu}_1,\tilde{\tau}_1}(x) - \mathcal{N}^{r,J}_{\tilde{\mu}_2,\tilde{\tau}_2}(x)\|_1 \leq \epsilon\,.$$

Proof This follows by a change of variables and Lemma 18. ■

Moreover, this rescaled parametrization naturally lends itself to approximation by a piecewise polynomial, namely, replace the standard normal Gaussian density function in Equation (6) with $\tilde{P}_\epsilon$. This is the piecewise polynomial that we will use to represent each individual component in the Gaussian mixture.

4.3. A parametrization scheme for k-GMMs

In the rest of this section, our parametrization will often be of the form described above. To distinguish this from the previous notation, for any $\theta \in \Theta_k$, and any set of k intervals $J_1, \ldots, J_k$, we will let $\theta^r \in \Theta_k$ denote the rescaled parameters so that if the i-th component in the mixture represented by θ has parameters w_i, μ_i, τ_i, then the i-th component in the mixture represented by θ^r has parameters $w_i, \tilde{\mu}_i, \tilde{\tau}_i$ so that $\mathcal{N}_{\mu_i,\tau_i} = \mathcal{N}^{r,J_i}_{\tilde{\mu}_i,\tilde{\tau}_i}$. Notice that the transformation between the original and the rescaled parameters is a linear transformation, and thus trivial to compute and to invert.

The final difficulty is that we do not know how many mixture components have associated interval J for $J \in \mathcal{I}$. To deal with this, our algorithm simply iterates over all possible allocations of the mixture components to intervals and returns the best one. There are $O(k)$ possible associated intervals J and k different components, so there are at most $k^{O(k)}$ different possible allocations. In this section, we will see how our parametrization works when we fix an allocation of the mixture components.

More formally, let $\mathcal{A}$ be the set of functions $v : [s] \to \mathbb{N}$ so that $\sum_{\ell=1}^{s} v(\ell) = k$. These will represent the number of components "allocated" to exist on the scale of each J_ℓ. For any $v \in \mathcal{A}$, define $\mathcal{I}_v$ to be the set of $I_\ell \in \mathcal{I}$ so that $v(\ell) \neq 0$.

Fix $\theta^r \in \Theta_k$ and $v \in \mathcal{A}$. Decompose θ^r into $(\theta^r_1, \ldots, \theta^r_s)$, where θ^r_ℓ contains the rescaled parameters with respect to J_ℓ for the $v(\ell)$ components allocated to interval J_ℓ (note that $v(\ell)$ may

be 0 in which case θ_ℓ is the empty set, i.e., corresponds to the parameters for no components). For any $1 \le \ell \le s$, let

$$\mathcal{M}^r_{\ell,\theta^r_\ell}(x) = \sum_i w_i \frac{\tilde{\tau}_i}{|I_\ell|/2} \mathcal{N}\left(\tilde{\tau}_i \cdot \frac{x - \mathrm{mid}(I_\ell)}{|I_\ell|/2} - \tilde{\mu}_i\right) ,$$

where i ranges over the components that θ_j corresponds to, and define $\mathcal{M}^r_{\theta^r,v}(x) = \sum_{\ell=1}^s \mathcal{M}^r_{\ell,\theta^r_\ell}(x)$. Similarly, define

$$P^r_{\epsilon,\ell,\theta^r_\ell}(x) = \sum_i w_i \frac{\tilde{\tau}_i}{|I_\ell|/2} \tilde{P}_\epsilon\left(\tilde{\tau}_i \cdot \frac{x - \mathrm{mid}(I_\ell)}{|J_\ell|/2} - \tilde{\mu}_i\right) ,$$

and define $P^r_{\epsilon,\theta^r,v}(x) = \sum_{\ell=1}^s P^r_{\epsilon,\ell,\theta^r_\ell}(x)$. Finally, for any v, define $\mathcal{P}^r_{\epsilon,v}$ to be the set of all such $P^r_{\epsilon,\theta,v}$.

We have:

Lemma 27 *For any $\theta^r \in \Theta_k$, we have*

$$\|\mathcal{M}^r_{\theta^r,v} - P^r_{\epsilon,\theta^r,v}\|_1 \le \epsilon .$$

This follows from roughly the same argument as in the proof of Lemma 8, and so we omit the proof.

We now finally have all the necessary language and tools to prove the following theorem:

Corollary 28 *Fix $2 \ge \epsilon > 0$. There is some allocation $v \in \mathcal{A}$ and a set of parameters $\theta^r \in \Theta_k$ so that $\tilde{\mu}_i \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$, $1/(8s) \le \tilde{\tau}_i \le \phi/2$, and $w_\ell \ge \epsilon/(2k)$ for all i. Moreover,*

$$\|f - \mathcal{M}^r_{\theta^r,v}\|_1 \le 19 \cdot \mathrm{OPT}_k + O(\epsilon) .$$

Proof Let $\theta^* \in \Theta_k$ be so that $\|f - \mathcal{M}_{\theta^*}\|_1 = \mathrm{OPT}_k$, and let $\mathcal{N}^*_\ell$ denote its ℓ-th component with parameters w^*_i, μ^*_i, and τ^*_i. Decompose $[k]$ into $S_{\mathrm{good}}(\theta^*), S_{\mathrm{bad}}(\theta^*)$ as in Lemma 23.

By the guarantees of the density estimation algorithm, we know that

$$\left\| \sum_\ell w^*_\ell \mathcal{N}_{\mu^*_\ell,\tau^*_\ell} - p_{\mathrm{dens}} \right\|_1 \le 5\mathrm{OPT}_k + \epsilon .$$

By Lemma 23, this implies that

$$5\mathrm{OPT}_k + \epsilon \ge \left\| \sum_{\ell \in S_{\mathrm{good}}(\theta^*)} w^*_\ell \mathcal{N}_{\mu^*_\ell,\tau^*_\ell} - p_{\mathrm{dens}} \right\|_1 + \frac{1}{2} \sum_{\ell \in S_{\mathrm{bad}}(\theta^*)} w_\ell - 2\epsilon ,$$

from which we may conclude the following two inequalities:

$$\left\| \sum_{\ell \in S_{\mathrm{good}}(\theta^*)} w^*_\ell \mathcal{N}_{\mu^*_\ell,\tau^*_\ell} - p_{\mathrm{dens}} \right\|_1 \le 5 \cdot \mathrm{OPT}_k + 3\epsilon, \tag{7}$$

$$\sum_{\ell \in S_{\mathrm{bad}}(\theta^*)} w^*_\ell \le 10 \cdot \mathrm{OPT}_k + 6\epsilon . \tag{8}$$

Let θ' be defined so that for all $\ell \in S_{\text{good}}(\theta^*)$, the means and variances of the ℓ-th component in θ' are μ_i^* and τ_i^*, and so that for all $\ell \in S_{\text{bad}}(\theta^*)$, the means of and variances of the ℓ-th component in θ' are arbitrary but so that the underlying Gaussian is admissible. Let the weights of the components in θ' be the same as the weights in θ^*.

Then we have

$$\begin{aligned}
\|M_{\theta'} - f\|_1 &= \left\| \sum_{\ell \in S_{\text{good}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell^*, \tau_\ell^*} + \sum_{\ell \in S_{\text{bad}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell', \tau_\ell'} - f \right\|_1 \\
&\le \left\| \sum_{\ell \in S_{\text{good}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell^*, \tau_\ell^*} - f \right\|_1 + \left\| \sum_{\ell \in S_{\text{bad}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell', \tau_\ell'} \right\|_1 \\
&= \left\| \sum_{\ell \in S_{\text{good}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell^*, \tau_\ell^*} - f \right\|_1 + \sum_{\ell \in S_{\text{bad}}(\theta^*)} w_\ell^* \\
&\le \left\| \sum_{\ell \in S_{\text{good}}(\theta^*)} w_\ell^* \mathcal{N}_{\mu_\ell^*, \tau_\ell^*} - p_{\text{dens}} \right\|_1 + \|f - p_{\text{dens}}\|_1 + \sum_{\ell \in S_{\text{bad}}(\theta^*)} w_\ell^* \\
&\le 19 \cdot \text{OPT}_k + O(\epsilon)
\end{aligned}$$

where the last line follows from Equation (7), the guarantee of the density estimation algorithm, and Equation (8).

For each $\ell \in [k]$, let $J_\ell \in \mathcal{I}$ denote the interval so that the ℓ-th component of θ' is admissible with respect to J_ℓ Let θ^r be the rescaling of θ' with respect to $J_1, \ldots, J_\ell$. Then by Lemma 25, θ^r satisfies that $\tilde{\mu}_i \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$ and $\sqrt{2\pi} \cdot \omega/(16s) \le \tilde{\tau}_i \le \phi/2$ for all i. Let $v \in \mathcal{A}$ be chosen so that $v(i)$ is the number of times that I_i appears in the sequence $J_1, \ldots, J_k$. Then $\mathcal{M}_{\theta'}$ and v satisfies all conditions in the lemma, except possibly that the weights may be too small.

Thus, let θ be the set of parameters whose means and precisions are exactly those of θ', but for which the weight of the ℓ-th component is defined to be $w_\ell = \max(\epsilon/(2k), w_\ell^*)$ for all $1 \le \ell \le k-1$ and $w_k = 1 - \sum_{\ell=1}^{k-1} w_\ell$. It is easy to see that $\theta \in \Theta_k$; moreover, $\|\mathcal{M}_\theta - \mathcal{M}_{\theta'}\|_1 \le \epsilon$. Then it is easy to see that θ and v together satisfy all the conditions of the lemma. ■

4.4. The full algorithm

At this point, we are finally ready to describe our algorithm LEARNGMM which agnostically and properly learns an arbitrary mixture of k Gaussians. Informally, our algorithm proceeds as follows. First, using ESTIMATE-DENSITY, we learn a p'_{dens} that with high probability is ϵ-close to the underlying distribution f in L_1-distance. Then, as before, we may rescale the entire problem so that the density estimate is supported on $[-1, 1]$. Call the rescaled density estimate p_{dens}.

As before, it suffices to find a k-GMM that is close to p_{dens} in $\mathcal{A}_K$-distance, for $K = 4k-1$. The following is a direct analog of Lemma 14. We omit its proof because its proof is almost identical to that of Lemma 14.

Lemma 29 *Let $\epsilon > 0, v \in \mathcal{A}$, $k \ge 2$, $\theta^r \in \Theta_k$, and $K = 4(k-1)+1$. Then we have*

$$0 \;\le\; \|p_{\text{dens}} - P^r_{\epsilon,\theta^r,v}\|_1 - \|p_{\text{dens}} - P^r_{\epsilon,\theta^r,v}\|_{\mathcal{A}_K} \;\le\; 8 \cdot \text{OPT}_k + O(\epsilon)\,.$$

Our algorithm enumerates over all $v \in \mathcal{A}$ and for each v finds a θ^r approximately minimizing

$$\|p_{\text{dens}} - P^r_{\epsilon,\theta^r,v}\|_{\mathcal{A}_K} .$$

Using the same binary search technique as before, we can transform this problem into $\log 1/\epsilon$ feasibility problems of the form

$$\|p_{\text{dens}} - P^r_{\epsilon,\theta^r,v}\|_{\mathcal{A}_K} < \eta . \tag{9}$$

Fix $v \in \mathcal{A}$, and recall $\mathcal{P}^r_{\epsilon,v}$ is the set of all polynomials of the form $P^r_{\epsilon,\theta^r,v}$. Let Θ_k^{valid} denote the set of $\theta^r \in \Theta_k$ so that $\tilde{\mu}_i \in [-\frac{2s\phi}{\omega}, \frac{2s\phi}{\omega}]$, $\sqrt{2\pi}\omega/(8s) \leq \tilde{\tau} \leq \phi/2$, and $w_i \geq \epsilon/(2k)$, for all i. For any $\theta^r \in \Theta_k^{\text{valid}}$, canonically identify it with $P^r_{\epsilon,\theta^r,v}$. By almost exactly the same arguments used in Section 3.5, it follows that the class $\mathcal{P}^r_{\epsilon,v}$, where $\theta \in \Theta_k^{\text{valid}}$, satisfies the conditions in Section 3.4, and that the system of polynomial equations $S_{K,p_{\text{dens}},\mathcal{P}^r_{\epsilon,v}}(\nu)$ has two levels of quantification (each with $O(k)$ bound variables), has $k^{O(k)}$ polynomial constraints, and has maximum degree $O(\log(1/\epsilon))$. Thus, we have

Corollary 30 *For any fixed ϵ, ν, and for $K = 4k-1$, we have that $S_{K,p_{\text{dens}},\mathcal{P}^r_{\epsilon,\nu},\Theta_k^{valid}}(\nu)$ encodes Equation (9) ranging over $\theta \in \Theta_k^{valid}$. Moreover, for all $\gamma, \lambda \geq 0$,*

$$\text{SOLVE-POLY-PROGRAM}(S_{K,p_{\text{dens}},\mathcal{P}^r_{\epsilon,\nu},\Theta_k^{valid}}(\nu), \lambda, \gamma)$$

runs in time

$$(k\log(1/\epsilon))^{O(k^4)} \log\log(3 + \frac{\gamma}{\lambda}) .$$

For each v, our algorithm then performs a binary search over η to find the smallest (up to constant factors) η so that Equation (9) is satisfiable for this v, and records both η_v, the smallest η for which Equation (9) is satisfiable for this v, and the output θ_v of the system of polynomial inequalities for this choice of η. We then return $\theta_{v'}$ so that the $\eta_{v'}$ is minimal over all $v \in \mathcal{A}$. The pseudocode for LEARNGMM is in Algorithm 2.

The following theorem is our main technical contribution:

Theorem 31 LEARNGMM(k, ϵ, δ) *takes $\widetilde{O}((k+\log 1/\delta)/\epsilon^2)$ samples from the unknown distribution with density f, runs in time*

$$\left(k\log\frac{1}{\epsilon}\right)^{O(k^4)} + \widetilde{O}\left(\frac{k}{\epsilon^2}\right) ,$$

and with probability $1-\delta$ returns a set of parameters $\theta \in \Theta_k$ so that $\|f - \mathcal{M}_\theta\|_1 \leq 58 \cdot \text{OPT} + \epsilon$.

Proof The sample complexity follows simply because ESTIMATE-DENSITY draws $\widetilde{O}((k+\log 1/\delta)/\epsilon^2)$ samples, and these are the only samples we ever use. The running time bound follows because $|\mathcal{A}| = k^{O(k)}$ and from Corollary 30. Thus it suffices to prove correctness.

Let θ be the parameters returned by the algorithm. It was found in some iteration for some $v \in \mathcal{A}$. Let v^*, θ^* be those which are guaranteed by Corollary 28. We have

$$\|p_{\text{dens}} - P^r_{\epsilon,\theta^*,v^*}\|_{\mathcal{A}_K} \leq \|p_{\text{dens}} - f\|_1 + \|f - \mathcal{M}^r_{\theta^*,v^*}\|_1 + \|\mathcal{M}^r_{\theta^*,v^*} - P^r_{\epsilon,\theta^*,v^*}\|_1 \leq 23\cdot\text{OPT}_k + O(\epsilon) .$$

By the above inequalities, the system of polynomial equations is feasible for $\eta \leq 46 \cdot \text{OPT}_k + O(\epsilon)$ in the iteration corresponding to v^* (Corollary 28 guarantees that the parameters θ^* are sufficiently

Algorithm 2 Algorithm for proper learning an arbitrary mixture of k Gaussians.

function LEARNGMM(k, ϵ, δ)
  ▷ Density estimation. Only this step draws samples.
  $p'_{\text{dens}} \leftarrow$ ESTIMATE-DENSITY(k, ϵ, δ)
  ▷ Rescaling
  ▷ p_{dens} is a rescaled and shifted version of p'_{dens} such that the support of p_{dens} is $[-1, 1]$.
  Let $p_{\text{dens}}(x) \stackrel{\text{def}}{=} p'_{\text{dens}}\left(\frac{2(x-\alpha)}{\beta-\alpha} - 1\right)$
  ▷ Fitting shape-restricted polynomials
  for $v \in \mathcal{A}$ **do**
    $\eta_v, \theta^r_v \leftarrow$ FINDFITGIVENALLOCATION(p_{dens}, v)
  Let θ so that $\theta^r = \theta^r_{v'}$ so that $\eta_{v'}$ is minimal over all η_v (breaking ties arbitrarily).
  ▷ Round weights back to be on the simplex
  for $i = 1, \ldots, k-1$ **do**
    $w_i \leftarrow w_i - \epsilon/2k$ (This guarantees that $\sum_{i=1}^{k-1} w_i \leq 1$; see analysis for details)
    If $w_i > 1$, set $w_i = 1$
  $w_k \leftarrow 1 - \sum_{i=1}^{k-1} w_i$
  ▷ Undo the scaling
  $w'_i \leftarrow w_i$
  $\mu'_i \leftarrow \frac{(\mu_i+1)(\beta-\alpha)}{2} + \alpha$
  $\tau'_i \leftarrow \frac{\tau_i}{\beta-\alpha}$
  return θ'

function FINDFITGIVENALLOCATION(p_{dens}, v)
  $\nu \leftarrow \epsilon$
  Let C_1 be a universal constant sufficiently small.
  Let $\lambda \leftarrow \min(C_1(\epsilon/(\phi k))^2, 1/16s, \epsilon/(4k))$
  ▷ This choice of precision provides robustness as needed by Lemma 26, and also ensures that all the weights and precisions returned must be non-negative.
  Let $\psi \leftarrow 6ks\phi/\omega + 3k\phi/2 + 1$
  ▷ By Corollary 28, this is a bound on how large any solution of the polynomial program can be.
  $\theta^r \leftarrow$ SOLVE-POLY-SYSTEM$(S_{p_{\text{dens}}, \mathcal{P}^r_{\epsilon,v}, \Theta_k^{\text{valid}}}(\nu), \lambda, \psi)$
  while θ^r is "NO-SOLUTION" **do**
    $\nu \leftarrow 2 \cdot \nu$
    $\theta^r \leftarrow$ SOLVE-POLY-SYSTEM$(S_{p_{\text{dens}}, \mathcal{P}^r_{\epsilon,v}, \Theta_k^{\text{valid}}}(\nu), \lambda, \psi)$
  return θ^r, ν

bounded). Hence, for some $\eta_{v^*} \leq \eta$, the algorithm finds some θ' so that there is some θ'' so that $\|\theta' - \theta''\|_2 \leq C_1(\epsilon/(\phi k))^2$, which satisfies $S_{p_{\text{dens}}, \mathcal{P}^r_{\epsilon,v^*}, \Theta_k^{\text{valid}}}(\nu_{v^*})$.

Let θ_1 be the set of parameters computed by the algorithm before rounding the weights back to the simplex (i.e. at Line 11). By our choice of precision in solving the polynomial program, (i.e. by our choice of λ on Line 24 of Algorithm 2), we know that the precisions of the returned mixture are

non-negative (so each component is a valid Gaussian). It was found in an iteration corresponding to some $v \in \mathcal{A}$, and there is some $\eta_v \le \eta_{v^*} \le 46 \cdot \mathrm{OPT}_k + O(\epsilon)$ and some θ_1' satisfying the system of polynomial equalities for v and η_v, so that $\|\theta_1 - \theta_1'\|_2 \le C_1(\epsilon/(\phi k))^2$. Let θ be the set of rescaled parameters obtained after rounding the weights of θ_1 back to the simplex. It is straightforward to check that $\theta \in \Theta_k$, and moreover, $\|\mathcal{M}^r_{\theta,v} - \mathcal{M}^r_{\theta_1',v}\|_1 \le 2\epsilon$, and so $\|P^r_{\epsilon,\theta,v} - P^r_{\epsilon,\theta_1',v}\|_1 \le O(\epsilon)$.

We therefore have

$$\begin{aligned}
\|f - M_\theta\|_1 &\le \|f - p_{\mathrm{dens}}\|_1 + \|p_{\mathrm{dens}} - P^r_{\epsilon,\theta,v}\|_1 + \|P^r_{\epsilon,\theta,v} - \mathcal{M}^r_{\epsilon,\theta,v}\|_1 \\
&\overset{(a)}{\le} 4 \cdot \mathrm{OPT} + \epsilon + 8 \cdot \mathrm{OPT} + O(\epsilon) + \|p_{\mathrm{dens}} - P^r_{\epsilon,\theta,v}\|_{\mathcal{A}_K} + \epsilon \\
&\overset{(b)}{\le} 12 \cdot \mathrm{OPT} + O(\epsilon) + \|p_{\mathrm{dens}} - P^r_{\epsilon,\theta_1',v}\|_{\mathcal{A}_K} \\
&\overset{(c)}{\le} 58 \cdot \mathrm{OPT} + O(\epsilon)\ ,
\end{aligned}$$

where (a) follows from Lemmas 29 and 27, (b) follows from the arguments above, and (c) follows since θ_1' satisfies the system of polynomial inequalities for $\eta_v \le 46 \cdot \mathrm{OPT}_k + O(\epsilon)$.

As a final step, we choose an internal ϵ' in our algorithm so that the $O(\epsilon')$ in the above guarantee becomes bounded by ϵ. This proves the desired approximation guarantee and completes the proof. ■

4.5. Further classes of distributions

Finally, we briefly show how to use our algorithm to properly learn other parametric classes of univariate distributions. Let $\mathcal{C}$ be a class of parametric distributions on the real line, parametrized by $\theta \in S$ for $S \subseteq \mathbb{R}^u$. For each θ, let $F_\theta \in \mathcal{C}$ denote the pdf of the distribution parametrized by θ in $\mathcal{C}$. To apply our algorithm in this setting, it suffices to show the following:

1. *(Simplicity of $\mathcal{C}$)* For any θ_1 and θ_2, the function $F_{\theta_1} - F_{\theta_2}$ has at most K zero crossings. In fact it also suffices if any two such functions have "essentially" K zero crossings.

2. *(Simplicity of S)* S is a semi-algebraic set.

3. *(Representation as a piecewise polynomial)* For each $\theta \in S$ and any $\epsilon > 0$, there is a a piecewise polynomial $P_{\epsilon,\theta}$ so that $\|P_{\epsilon,\theta} - F_\theta\|_1 \le \epsilon$. Moreover, the map $(x, \theta) \mapsto P_{\epsilon,\theta}(x)$ is jointly polynomial in x and θ at any point so that x is not at a breakpoint of $P_{\epsilon,\theta}$. Finally, the breakpoints of $P_{\epsilon,\theta}$ also depend polynomially on θ.

4. *(Robustness of the Parametrization)* There is some robust parametrization so that we may assume that all "plausible candidate" parameters are $\le 2^{\mathrm{poly}(1/\epsilon)}$, and moreover, if $\|\theta_1 - \theta_2\| \le 2^{-\mathrm{poly}(1/\epsilon)}$, then $\|F_{\theta_1} - F_{\theta_2}\| \le \epsilon$.

Assuming $\mathcal{C}$ satisfies these conditions, our techniques immediately apply. In this paper, we do not attempt to catalog classes of distributions which satisfy these properties. However, we believe such classes are often natural and interesting. We give evidence for this below, where we show that our framework produces proper and agnostic learning algorithms for mixtures of two more types of simple distributions. The resulting algorithms are both sample optimal (up to log factors) and have nearly-linear running time.

4.5.1. LEARNING MIXTURES OF SIMPLE DISTRIBUTION

As a brief demonstration of the generality of our technique, we show that our techniques give proper and agnostic learning algorithms for mixtures of k exponential distributions and Laplace distributions (in addition to mixtures of k Gaussians) which are *nearly-sample optimal*, and run in time which is *nearly-linear* in the number of samples drawn, for any constant k.

We now sketch a proof of correctness for both classes mentioned above. In general, the robustness condition is arguably the most difficult to verify of the four conditions required. However, it can be verified that for mixtures of simple distributions with reasonable smoothness conditions the appropriate modification of the parametrization we developed in Section 4 will suffice. Thus, for the classes of distributions mentioned, it suffices to demonstrate that they satisfy conditions (1) to (3).

Condition 1: It follows from the work of Tossavainen (2006) that the difference of k exponential distributions or k Laplace distributions has at most $2k$ zero crossings.

Condition 2: This holds trivially for the class of mixtures of exponential distributions. We need a bit of care to demonstrate this condition for Laplace distributions since a Laplace distribution with parameters μ, b has the form

$$\frac{1}{2b}e^{-|x-\mu|/b}$$

and thus the Taylor series is not a polynomial in x or the parameters. However, we may sidestep this issue by simply introducing a variable y in the polynomial program which is defined to be $y = |x - \mu|$.

Condition 3: It can easily be shown that a truncated degree $O(\log 1/\epsilon)$ Taylor expansion (as of the form we use for learning k-GMMs) suffices to approximate a single exponential or Laplace distribution, and hence a $O(k)$-piecewise degree $O(\log 1/\epsilon)$ polynomial suffices to approximate a mixture of k exponential or Laplace distributions up to L_1-distance ϵ.

Thus for both of these classes, the sample complexity of our algorithm is $\widetilde{O}(k/\epsilon^2)$, and its running time is

$$\left(k \log \frac{1}{\epsilon}\right)^{O(k^4)} + \widetilde{O}\left(\frac{k}{\epsilon^2}\right) ,$$

similar to the algorithm for learning k-GMMs. As for k-GMMs, this sample complexity is nearly optimal, and the running time is nearly-linear in the number of samples drawn, if k is constant.

5. Our high-dimensional algorithm

In this section, we extend our univariate proper learning algorithm to learn mixtures of spherical Gaussians in high dimensions. At a high level, our algorithm proceeds as follows:

1. We approximate the covariance of each component to low accuracy.
2. We employ a recursive spectral algorithm to produce a clustering of the samples with the following guarantee: with high probability, all samples from the same mixture component are assigned to the same cluster. Moreover, the component means in each cluster differ by at most $\widetilde{O}(k)$. For each cluster, we also estimate the total weight of all mixture components in the cluster to high accuracy. We then consider each cluster as a separate subproblem.

3. In each cluster, we compute an approximate PCA of the sample covariance matrix to produce a k-dimensional subspace S which is approximately the span of the component means. Moreover, we use an approximation of the $(k+1)$st eigenvalue of the sample covariance matrix as an estimate for the component precision $\tau = 1/\sigma$.

4. Let k' be the number of components in a cluster. We then find a set of $(\mathrm{poly}(k'))^{O(k')}$ "good" directions in this subspace and estimate the density of the unknown distribution on each of these directions. Next, we run our univariate learning algorithm *simultaneously* on all of these directions to find a *single* mixture of k' spherical Gaussians (with common covariance) in $\mathbb{R}^{k'}$ satisfying the following property: on each direction we consider, the projection of this mixture is close (in total variation distance) to the projection of the unknown distribution on this direction. We output this mixture embedded in $\mathbb{R}^d$.

5. Finally, we combine the estimates in each cluster and output the resulting mixture.

There is an additional twist here: a priori, we do not know how many mixture components each cluster contains. We get around this issue by enumerating over all possible ways to split the components into each cluster, producing $2^{O(k)}$ different proper hypotheses. We then perform hypothesis selection over this set.

Steps 1 to 3 are essentially the algorithm presented in Acharya et al. (2014), but we use faster approximate PCA techniques and better concentration bounds. Our exposition of Steps 1 to 3 primarily serves to clarify and strengthen the argument presented in Acharya et al. (2014), and also to show that the approximate PCA guarantees suffice for our purposes. Our main contribution is Step 4 in which we extend our univariate proper learning algorithm to the k-dimensional setting. The main challenge is to show that *univariate* density estimates suffice for proper learning in the k-dimensional space.

5.1. Preliminaries

We adapt our notation from the previous sections to the high-dimensional setting. In the following, we use d as the full ambient dimension, i.e., our samples are d-dimensional real vectors. We denote vectors with lower case letters in bold face and let $\|\boldsymbol{u}\|_2$ denote the Euclidean norm of a vector $\boldsymbol{u}$. Moreover, we let $\|\boldsymbol{A}\|_{\mathrm{op}}$ denote the ℓ_2 operator norm (or spectral norm) of a matrix $\boldsymbol{A}$, and we use upper case letters in bold face for matrices. Since we focus on mixtures of spherical Gaussians, it suffices to quantify the normal pdf with a mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and a scalar precision parameter $\tau = 1/\sigma$, which gives:

$$\mathcal{N}_{\boldsymbol{\mu},\tau}(\boldsymbol{x}) = (2\pi)^{-d/2}\tau^d \cdot e^{-\frac{1}{2}\tau^2\|\boldsymbol{x}-\boldsymbol{\mu}\|_2^2} .$$

As before, we denote the set of valid k-GMM parameters with Θ_k, where a valid parameter vector $\theta \in \Theta_k$ now satisfies $\tau_i = \tau$ for all $i \in [k]$. We associate these parameters with the d-dimensional mixture of Gaussians

$$\mathcal{M}_\theta(\boldsymbol{x}) \;=\; \sum_{i=1}^{k} w_i \cdot \mathcal{N}_{\boldsymbol{\mu}_i,\tau_i}(\boldsymbol{x}) \; .$$

If $\boldsymbol{v} \in \mathbb{R}^d$ is a unit vector, we let $\boldsymbol{v} \cdot \theta$ denote the set of parameters of the univariate marginal distribution induced by projecting the distribution $\mathcal{M}_\theta$ along $\boldsymbol{v}$. That is, $\boldsymbol{v} \cdot \theta$ is the set of parameters with the same weights w_j and precision τ_j as the set of parameters θ, but with means $\boldsymbol{v} \cdot \boldsymbol{\mu}_j$.

Formally, we consider the following problem: given m independent samples $\boldsymbol{x}_1, \dots, \boldsymbol{x}_m \in \mathbb{R}^d$ drawn from the distribution $\mathcal{M}_{\theta^*}$, where θ^* satisfies $\tau_i^* = \tau_j^*$ for all i, j, our goal is to recover a set of parameters $\theta \in \Theta_k$ so that we have the L_1-norm guarantee $\|\mathcal{M}_{\theta^*} - \mathcal{M}_\theta\|_1 \leq O(\epsilon)$. We also provide agnostic guarantees for our algorithm, building upont the recent work of Diakonikolas et al. (2016a).

We are generally interested in the regime where the parameter k is constant and the parameters d and $1/\epsilon$ are growing. In particular, we will assume that $d > k$, that $1/\epsilon > k$, and that $d \geq O(\log(k/\delta))$, where δ is the failure probability of our algorithm.

In the following arguments, we often need to find the top k eigenvectors of our data matrix $\boldsymbol{A} \in \mathbb{R}^{d \times m}$ given by the samples $\boldsymbol{x}_i$. In all instances, it turns out that an *approximate* notion of the top k eigenvectors suffices for our algorithm, which allows us to use computationally more efficient algorithms. In particular, we use the following algorithm, which is essentially the randomized block Krylov method analyzed in Musco and Musco (2015) applied to the matrix $\boldsymbol{A}\boldsymbol{A}^T$:

Theorem 32 (Musco and Musco (2015)) *Fix $\epsilon > 0$. Let $\boldsymbol{A} \in \mathbb{R}^{d \times n}$ be an arbitrary matrix. Then there is an algorithm $\text{APPROXPCA}(\boldsymbol{A}, k, \epsilon)$ that runs in time $O(kdn/\sqrt{\epsilon})$ and returns an orthonormal basis $V = \{\boldsymbol{v}_1, \dots, \boldsymbol{v}_k\}$, or equivalently, a projector Π_V onto span(V), so that*

$$\|\boldsymbol{A}\boldsymbol{A}^T - \Pi_V \boldsymbol{A}\boldsymbol{A}^T\|_2 \;\leq\; (1+\epsilon)\sigma_{k+1} \;,$$

where σ_{k+1} is the $(k+1)$st largest singular value of $\boldsymbol{A}$. Moreover, if $\boldsymbol{u}_1, \dots, \boldsymbol{u}_d$ are the left singular vectors of $\boldsymbol{A}$ sorted in decreasing order of their corresponding singular values, we have

$$|\boldsymbol{v}_i \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{v}_i - \boldsymbol{u}_i \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{u}_i| \;\leq\; \epsilon\sigma_{k+1} \;.$$

5.1.1. INFORMATION THEORETIC TOOLS

KL divergence and Pinsker's inequality The KL divergence (or relative entropy) is a well-studied "measure" of distance between probability distributions. In our setting, it can be defined as follows.

Definition 33 (KL divergence) *Let P, Q be two probability density functions over $\mathbb{R}^d$. Then*

$$D_{KL}(P||Q) = \int_{\mathbb{R}^d} \log \frac{P(x)}{Q(x)} P(x)\,\mathrm{d}x \;.$$

In our analysis here, we are mainly interested in the KL divergence because it allows us to establish upper bounds on the total variation distance between two distributions:

Theorem 34 (Pinsker's inequality, see e.g. Tsybakov (2008)) *Let P, Q be two probability density functions over $\mathbb{R}^d$. Then*

$$\|P - Q\|_1 \leq \sqrt{2 D_{KL}(P||Q)} \;.$$

Moreover, the KL divergence between two multivariate Gaussians has a convenient closed form:

Fact 35 *Let P, Q be normal distributions with means $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2$ and covariance matrices $\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2$. Assume that $\det(\boldsymbol{\Sigma}_1), \det(\boldsymbol{\Sigma}_2) > 0$. Then*

$$d_{KL}(P||Q) = \frac{1}{2}\left(tr(\boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1) - d + (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1)^T \boldsymbol{\Sigma}_2 (\boldsymbol{\mu}_2 - \boldsymbol{\mu}_1) + \log \frac{|\boldsymbol{\Sigma}_1|}{|\boldsymbol{\Sigma}_0|} \right) \;.$$

Together, these two statements imply the following:

Corollary 36 *Let $P = \mathcal{N}_{\boldsymbol{\mu}_1, 1}$ and $Q = \mathcal{N}_{\boldsymbol{\mu}_2, 1}$. Then $\|P - Q\|_1 \leq O(\|\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2\|_2)$.*

Data Processing Inequality We require the following well-known version of the Data Processing Inequality:

Theorem 37 (Data Processing Inequality, see e.g. Cover and Thomas (2006)) *Let D, D' be distributions on some abstract measurable space, and let F be a random function. Define $F(D)$ to be the distribution given by first drawing a random sample x from D and then outputting $F(x)$ (where F may be randomly chosen from the random family). We adopt the same definition for $F(D')$. Then*

$$d_{TV}(F(D), F(D')) \;\leq\; d_{TV}(D, D') \;.$$

For any subspace S and probability distribution D, let D_S denote the projection of D onto S, i.e., to draw a sample from D_S, we first draw a sample from D and then project it onto S. Letting F be the function which projects samples from D onto S, so that $D_S = F(D)$, we see that this is a special case of the setting in the Data Processing inequality. Hence, as a corollary, we obtain:

Corollary 38 *Let D, D' be two distributions, and let S be any subspace. Then*

$$\|D_S - D'_S\|_1 \;\leq\; \|D - D'\|_1 \;.$$

5.1.2. CONCENTRATION BOUNDS

In this subsection, we establish concentration results that we will require in the future. We require the following preliminaries. The first bounds the largest deviation of any single point from a Gaussian:

Fact 39 (Folklore) *Fix $\delta > 0$. Let $y_1, \ldots, y_n$ be n independent samples from the standard univariate Gaussian distribution. Then, with probability $1 - \delta$, we have $\sup_{i \in [n]} |y_i| \leq \sqrt{\log(n/\delta)}$.*

The second bounds the deviation of the mean of the samples:

Fact 40 (Folklore) *Fix $\epsilon, \delta > 0$. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be $n = O\left(\frac{d + \log(1/\delta)}{\epsilon^2}\right)$ independent samples from $\mathcal{N}_{\mathbf{0},1}$. Then with probability $1 - \delta$,*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i \right\|_2 \;\leq\; \epsilon \;.$$

Theorem 41 (Corollary 5.50 in Vershynin (2010)) *Fix $\epsilon, \delta > 0$. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be $n = O\left(\frac{d + \log(1/\delta)}{\epsilon^2}\right)$ independent samples from $\mathcal{N}_{\mathbf{0},1}$. Then with probability $1 - \delta$,*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{y}_i \boldsymbol{y}_i^T - \boldsymbol{I} \right\|_{\mathrm{op}} \;\leq\; \epsilon \;.$$

This allows us to show the following:

Corollary 42 *Fix $\epsilon, \delta > 0$. Let $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n$ be $n = O\left(\frac{d + \log(1/\delta)}{\epsilon^2}\right)$ independent samples from $\mathcal{N}_{\boldsymbol{\mu},1}$, and let $\boldsymbol{\mu} \in \mathbb{R}^d$ be arbitrary. Then with probability $1 - \delta$,*

$$\left\| \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{y}_i + \boldsymbol{\mu})(\boldsymbol{y}_i + \boldsymbol{\mu})^T - (\boldsymbol{I} + \boldsymbol{\mu}\boldsymbol{\mu}^T) \right\|_{\mathrm{op}} \;\leq\; O(\epsilon(1 + \|\boldsymbol{\mu}\|_2)) \;.$$

Proof By expanding out the LHS, we get that

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i+\boldsymbol{\mu})(\boldsymbol{y}_i+\boldsymbol{\mu})^T-(\boldsymbol{I}+\boldsymbol{\mu}\boldsymbol{\mu}^T) \;=\; \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{y}_i^T-\boldsymbol{I}+\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{\mu}^T+\boldsymbol{\mu}^T\left(\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\right)\,.$$

We then have

$$\begin{aligned}\left\|\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{y}_i+\boldsymbol{\mu})(\boldsymbol{y}_i+\boldsymbol{\mu})^T-(\boldsymbol{I}+\boldsymbol{\mu}\boldsymbol{\mu}^T)\right\|_{\mathrm{op}} &\le \left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\boldsymbol{y}_i^T-\boldsymbol{I}\right\|_{\mathrm{op}}+2\|\boldsymbol{\mu}\|_2\left\|\frac{1}{n}\sum_{i=1}^{n}\boldsymbol{y}_i\right\|_2\\ &\le \epsilon+2\epsilon\|\boldsymbol{\mu}\|_2\,,\end{aligned}$$

where the last line follows from Fact 40 and Theorem 41. ■

We now give bounds for the rate of convergence of the sample mean and covariance for a mixture of Gaussians $\mathcal{M}_\theta$ with shared covariance $\boldsymbol{I}$. In the following, let $\boldsymbol{x}_1,\ldots,\boldsymbol{x}_n$ be n samples from $\mathcal{M}_\theta$. For component j, let S_j denote the subset of the samples that were drawn from component j. We let $\widehat{w}_j=|S_j|/n$ be the empirical mixing weights, and we let $\widehat{\boldsymbol{\mu}}=\frac{1}{n}\sum\boldsymbol{x}_i$ be the empirical mean. Furthermore, we let $\widetilde{\boldsymbol{\mu}}=\sum\widehat{w}_j\boldsymbol{\mu}_j$ and

$$\widetilde{\boldsymbol{C}} \;=\; \sigma^2\boldsymbol{I}+\sum_{j=1}^{k}\widehat{w}_j(\boldsymbol{\mu}_j-\widetilde{\boldsymbol{\mu}})(\boldsymbol{\mu}_j-\widetilde{\boldsymbol{\mu}})^T \tag{10}$$

be the mean and covariance with the empirical mixing weights instead of the true mixing weights. We also define

$$\gamma \;=\; \max_j \widehat{w}_j\|\boldsymbol{\mu}_j-\widetilde{\boldsymbol{\mu}}\|_2^2\,, \tag{11}$$

which will be an important parameter for us later.

In the concentration argument below, we ignore all components with $\widehat{w}_j=0$ because we often divide by $\widehat{w}_j$. It is easy to see that this does not affect any guarantees we prove in this subsection.

We first show that the empirical mean is close to $\widetilde{\boldsymbol{\mu}}$:

Lemma 43 *Fix $\epsilon,\delta>0$, and let $n=O(\frac{d+\log(k/\delta)}{\epsilon^2})$. With probability $1-\delta$, we have*

$$\|\widetilde{\boldsymbol{\mu}}-\widehat{\boldsymbol{\mu}}\|_2 \;\le\; O(k^{1/2}\epsilon)\,.$$

Proof By Fact 40, we know that with probability $1-\delta/k$,

$$\left\|\boldsymbol{\mu}_j-\frac{1}{|S_j|}\sum_{i\in S_j}\boldsymbol{x}_i\right\|_2 \;\le\; O\left(\frac{\epsilon}{\widehat{w}_j^{1/2}}\right)\,.$$

We then have that with probability $1-\delta$,

$$\begin{aligned}\left\|\boldsymbol{\mu}' - \frac{1}{n}\sum_{i=1}^{n}\boldsymbol{x}_i\right\|_2 &= \left\|\sum_{j=1}^{k}\left(\widehat{w}_j\boldsymbol{\mu}_j - \widehat{w}_j\frac{1}{|S_j|}\sum_{i\in S_j}\boldsymbol{x}_j\right)\right\|_2 \\ &\le \sum_{j=1}^{k}\widehat{w}_j\left\|\boldsymbol{\mu}_j - \frac{1}{|S_j|}\sum_{i\in S_j}\boldsymbol{x}_j\right\|_2 \\ &\le \sum_{j=1}^{k}O(\widehat{w}_j^{1/2}\epsilon) \\ &\le O(k^{1/2}\epsilon)\ ,\end{aligned}$$

since $\sum_j \widehat{w}_j = 1$. ■

This immediately leads to the following corollary:

Corollary 44 *Fix ϵ, δ, and let $n = O(\frac{d+\log(k/\delta)}{\epsilon^2})$. With probability $1-O(\delta)$, for all j, we have*

$$\left|\|\boldsymbol{\mu}_i - \widehat{\boldsymbol{\mu}}\|_2 - \|\boldsymbol{\mu}_i - \boldsymbol{\mu}'\|_2\right| \le O(k^{1/2}\epsilon)\ .$$

Now we turn to bounding the covariance:

Lemma 45 *Fix $\epsilon, \delta > 0$, and let $n = O(\frac{d+\log(k/\delta)}{\epsilon^2})$. With probability $1-O(\delta)$, we have that*

$$\left\|\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T - \widetilde{\boldsymbol{C}}\right\|_{\mathrm{op}} \le O(k^{1/2}\epsilon + \epsilon k\gamma^{1/2} + k\epsilon^2)\ ,$$

where $\widetilde{\boldsymbol{C}}$ is as defined in Equation (10).

Proof

We write

$$\frac{1}{n}\sum_{i=1}^{n}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T = \sum_{j=1}^{k}\widehat{w}_j\frac{1}{|S_j|}\sum_{i\in S_j}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T\ .$$

Note that

$$\begin{aligned}\sum_{i\in S_j}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T &= \sum_{i\in S_j}(\boldsymbol{x}_i - \boldsymbol{\mu}_j + \boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \boldsymbol{\mu}_j + \boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})^T \\ &= \sum_{i\in S_j}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T + (\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})\sum_{i\in S_j}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \\ &\quad + \left(\sum_{i\in S_j}(\boldsymbol{x}_i - \boldsymbol{\mu}_j)\right)(\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})^T + |S_j|(\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})(\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})^T\ .\end{aligned}$$

By Fact 40 and Theorem 41, the following inequalities hold with probability $1 - O(\delta)$:

$$\left\| \frac{1}{|S_j|} \sum_{i \in S_j} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T - I \right\|_{\mathrm{op}} \leq \frac{\epsilon}{\widehat{w}_j^{1/2}} , \qquad \text{and}$$

$$\left\| \frac{1}{|S_j|} (\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}) \sum_{i \in S_j} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T \right\|_2 \leq \|\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}\|_2 \frac{\epsilon}{\widehat{w}_j^{1/2}}$$

and so, putting things together via triangle inequalities, we have

$$\begin{aligned}
\left\| \frac{1}{n} \sum_{i=1}^{n} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}})^T - \widetilde{\boldsymbol{C}} \right\|_{\mathrm{op}} &\leq \sum_{j=1}^{k} \frac{|S_j|}{n} \left(\frac{\epsilon}{\widehat{w}_j^{1/2}} + 2\|\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}\|_2 \frac{\epsilon}{\widehat{w}_j^{1/2}} \right) \\
&\leq k^{1/2}\epsilon + 2\epsilon \sum_{j=1}^{k} \widehat{w}_j^{1/2} \|\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}\|_2 \\
&\leq k^{1/2}\epsilon + 2\epsilon \sum_{j=1}^{k} \widehat{w}_j^{1/2} \left(\|\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}\|_2 + \|\widetilde{\boldsymbol{\mu}} - \widehat{\boldsymbol{\mu}}\|_2 \right) \\
&\leq k^{1/2}\epsilon + 2\epsilon \sum_{j=1}^{k} \widehat{w}_j^{1/2} \left(\|\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}\|_2 + O(k^{1/2}\epsilon) \right) \\
&= O(k^{1/2}\epsilon + \epsilon k \gamma^{1/2} + k\epsilon^2) ,
\end{aligned}$$

where the second to last line follows from Corollary 44 and the last line follows from the definition of γ (Equation (11)). ■

5.2. The algorithm

We now formally define our algorithm LEARNMULTIVARIATEGMM in Algorithm 3. To prove the desired proper learning guarantee, we analyze the distortion incurred when projecting high dimensional GMMs along a line. In particular, we are interested in the L_1-distance between two high dimensional mixtures, and the L_1-distance between the same two mixtures projected onto a line. By classical arguments (the Data Processing Inequality), it is clear that the L_1-distance cannot increase due to the projection. For our arguments, we need the opposite direction, i.e., we want to show that univariate projection does not *decrease* the total variation distance too much. To this effect, we define the following quantity.

Definition 46 (Projected separation) *Let $\epsilon > 0$ be the L_1-distance separation in the high-dimensional space, let $k \in \mathbb{N}$ be the number of components per mixture, and let $V \subset S^{k-1}$ be a given set of direction in $\mathbb{R}^k$. Then we define the projected separation $\zeta \in \mathbb{R}$ to be*

$$\zeta(\epsilon, k, V) = \inf_{\theta', \theta'' : \|\mathcal{M}_{\theta'} - \mathcal{M}_{\theta''}\|_1 \geq \epsilon} \max_{\boldsymbol{v} \in V} \|\mathcal{M}_{\boldsymbol{v} \cdot \theta'} - \mathcal{M}_{\boldsymbol{v} \cdot \theta''}\|_1 , \tag{12}$$

where the minimum is taken over all k-dimensional mixtures of Gaussians with parameters θ', θ'' so that all covariances in all mixture components are $\sigma^2 \boldsymbol{I}$ for some common σ.

The projected separation $\zeta(\epsilon, k, V)$ measures how much the distance between any two mixtures of k Gaussians in $\mathbb{R}^k$ with the same spherical covariance can decrease if we only consider directions in V. When the parameters are well-understood, we sometimes simply write ζ for this quantity.

A priori, it is not clear that the projected separation ζ is non-zero. Indeed, for a small set V, the separation ζ can in general be zero. However, it is not too hard to show that there is a set of directions $V_{\epsilon,k}$ of size $|V_{\epsilon,k}| = (1/\epsilon)^{O(k)}$ so that $\zeta(\epsilon, k, V_{\epsilon,k}) = \Omega(\epsilon)$ (take V to be an ϵ-net for S^{k-1}). With this choice of V, our algorithm roughly recovers the sample and time guarantees of the prior work Acharya et al. (2014). However, we believe that the bound on the cardinality of V can be improved significantly: Conjecture 1 implies that for all k, there is a set of directions V_k with size $|V_k| = (\mathrm{poly}(k))^{O(k)}$ so that $\zeta(\epsilon, k, V_k) = \Omega_k(\epsilon)$. A crucial aspect of this conjecture is that the cardinality of V_k is *independent* of ϵ. Such a set of directions V_k would allow us to achieve a significantly better time complexity than Acharya et al. (2014) for any fixed k. In particular, we would separate the exponential dependence between $1/\epsilon$ and k. In Subsection 9, we show that in the case of two mixture components (i.e., $k = 2$), we can indeed give a set of directions V such that the cardinality $|V|$ is independent of ϵ.

The overall guarantee of our algorithm is the following:

Theorem 47 LEARNMULTIVARIATEGMM(k, ϵ, δ, V) *requires*

$$N = \widetilde{O}_k\left(\frac{d}{\epsilon^4} + \frac{d + \log(|V|/\delta)}{\zeta^2}\right)$$

samples and time

$$\widetilde{O}_k\left(\frac{|V|}{\zeta^2} + d^2\left(\frac{1}{\epsilon^4} + \frac{1}{\zeta^2}\right) \cdot \min\left(\frac{1}{\epsilon}, \sqrt{\frac{1}{\zeta}}\right) + \left(|V| \log \frac{1}{\zeta}\right)^{O(|V|^2)}\right)$$

and with probability $1 - \delta$*, it returns a mixture of* k *Gaussians* $\mathcal{M}_\theta$ *so that* $\|\mathcal{M}_\theta - \mathcal{M}_{\theta^*}\|_1 \leq \epsilon$.

We state our dependence on k explicitly in Section 10 but suppress it here for clarity. An important consequence of Theorem 47 is the following: assuming Conjecture 1, our algorithm has both good time and sample complexity:

Corollary 48 *Assuming Conjecture 1, the sample complexity of our algorithm* LEARNMULTIVARIATEGMM *is* $\widetilde{O}_k\left(\frac{d}{\epsilon^4}\right)$ *and its running time is* $\widetilde{O}_k\left(\frac{d^2}{\epsilon^5}\right)$.

We believe that Conjecture 1 holds for all values of k and give numerical evidence in Section 11. Proving Conjecture 1 is a promising direction for future work. Here, we prove a version of Conjecture 1 for the $k = 2$ case with $\zeta = \epsilon^3$, which directly gives the following result:

Corollary 49 *There is a set of directions* V *of constant size that can also be computed in constant time so that the following guarantee holds: the algorithm* LEARNMULTIVARIATEGMM$(2, \epsilon, \delta, V)$ *returns a mixture of two Gaussians* $\mathcal{M}_\theta$ *so that* $\|\mathcal{M}_\theta - \mathcal{M}_{\theta^*}\|_1 \leq \epsilon$ *with probability* $1 - \delta$*. Moreover, the algorithm requires* $\widetilde{O}\left(\frac{d \log(1/\delta)}{\epsilon^6}\right)$ *samples and runs in time* $\widetilde{O}\left(\frac{d^2}{\epsilon^{7.5}}\right)$.

Corollary 49 gives the best known time complexity for proper learning of a high-dimensional 2-GMM with shared spherical covariance.

Algorithm 3 Our algorithm for learning a mixture of multivariate Gaussians with the same covariance.

function LEARNMULTIVARIATEGMM(ϵ, δ, V)
  ▷ Step 1: get a coarse estimate of the shared variance σ
  $\sigma_1 = \text{COARSEESTIMATESIGMA}(k)$.
  In all later steps, we divide all samples by σ_1.
  Let $S^1, \dots, S^k$ be k sets of independent samples, each containing n samples, where

$$n = \widetilde{O}\left(\frac{dk^9 \log(k/\delta)}{\epsilon^4} + \frac{dk^{7/2} \log(|V|/\delta)}{\zeta^2}\right).$$

  ▷ Step 2: recursive spectral projection clustering
  Run $\text{RECURSIVESPECTRALPROJECTION}(S^1, \dots, S^k, \delta)$.
  Let t be the iteration that RECURSIVESPECTRALPROJECTION returns on.
  Let $\mathcal{T}_t$ be the returned clustering tree.
  Let $S^t_1, \dots, S^t_{k'}$ be the partition of S^t induced by $\mathcal{T}$.
  for cluster C_r **do**
    ▷ Step 3: Find an approximation for the subspace spanned by the means.
    ▷ We also obtain a more accurate estimate of the covariance.
    Let $\Pi_r, \widehat{\sigma}_r = \text{FINDAPPROXSUBSPACEANDCOVARIANCE}(k, S^t, \epsilon, \delta, \zeta, \mathcal{T}_t, r)$
    If the number of samples in C_r is at least a $O(1/k)$ fraction of all the samples, let $\widehat{\sigma} = \widehat{\sigma}_r$.
    for $\ell = 1, \dots, k$ **do**
      ▷ Step 4: Find a proper estimate for the current cluster.
      ▷ This step relies on density estimation and a system of polynomial inequalities.
      $\theta_{r,\ell} \leftarrow \text{FITPOLYPROGRAMMULTIVARIATE}(\ell, \Pi_r, S^t_r, \widehat{\sigma}, \epsilon, \delta)$.
  ▷ Hypothesis selection since we do not know how many components are in each cluster
  for all sets of positive integers $\beta_1, \dots, \beta_\ell$ so that $\sum \beta_r = k$ **do**
    Form a hypothesis $\sum \widehat{w}_r \mathcal{M}_{\theta_{r,\beta_\ell}}$.
  Perform hypothesis selection on all hypotheses formed this way.
  return the winning hypothesis.

6. Spectral projection clustering

In this section, we describe our algorithm for clustering via recursive spectral projections. As a first step, we obtain an estimate for the common covariance.

6.1. Finding a rough estimate for the covariance of each component

Similar to Acharya et al. (2014), we use an algorithm $\text{COARSEESTIMATESIGMA}(k)$ that works as follows: take $n+1$ samples $\boldsymbol{y}_1, \dots, \boldsymbol{y}_{n+1}$ and output $\sigma_1 = \min_{i,j} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2$. This algorithm goes back to at least Bishop (1995) (see also Dasgupta and Schulman (2007)).

Lemma 50 (Lemma 4 in Acharya et al. (2014)) *If $d \geq O(\log(n/\delta))$, then with probability $1-\delta$, we have $|\sigma_1^2 - \sigma^2| \leq O(\sigma^2)$.*

Thus by dividing all samples by σ_1 and working with the distribution that results from this transformation, we may assume that $\sigma = \Theta(1)$. We make this assumption for the rest of this section. It can be easily verified that all the concentration inequalities we proved previously still hold when $\sigma = \Theta(1)$, just with possibly different constant factors.

6.2. The projection clustering algorithm

In this section, we describe our algorithm for clustering via recursive spectral projections. The most basic objects we will use to cluster are what we call *clustering directions*.

Definition 51 *A clustering direction for a k-GMM $\mathcal{M}$ with error δ is a unit vector $\boldsymbol{v} \in \mathbb{R}^d$ such that there exists a proper, nonempty subset of components $C \subsetneq [k]$ with the following property: With probability $1-\delta$ over a draw $\boldsymbol{x} \sim \mathcal{M}$, we have that the sample $\boldsymbol{x}$ was drawn from the components in C if and only if $\boldsymbol{v}^T\boldsymbol{x} > 0$.*

Thus, a clustering direction gives a single linear test that implies a non-trivial clustering for the k-GMM $\mathcal{M}$. To cluster further, we may then take additional samples, use this test to partition them into two clusters, and then recursively partition the two clusters with new clustering directions. We formalize this process as *clustering trees*.

Definition 52 *A* clustering tree *is a binary tree so that no node has only one children. Each non-leaf node is labeled with a linear function* $\mathrm{label} : \mathbb{R}^d \to \mathbb{R}$. *Given any point $\boldsymbol{x} \in \mathbb{R}^d$, we define its associated leaf node to be the leaf node that one arrives at by recursively navigating the tree starting from the root, at each step traversing to the left child if* $\mathrm{label}(\boldsymbol{x}) < 0$, *and traversing to the right child otherwise.*

Intuitively, a clustering tree partitions $\mathbb{R}^d$ into intersections of halfspaces, and we can use this partition to cluster points from mixtures of Gaussians:

Definition 53 *Fix a mixture of k Gaussians $\mathcal{M}$, and let $\mathcal{T}$ be a clustering tree with $k' \leq k$ leaves. We say that that the tree $\mathcal{T}$ is a* valid *clustering tree for $\mathcal{M}$ with error probability δ if there is a partition $\{C_1, \ldots, C_{k'}\}$ of the components of $\mathcal{M}$ such that the following property holds: if $\boldsymbol{x}$ is a sample drawn from any component in C_i, then its associated leaf node is the i-th leaf node with probability at least $1-\delta$. Given a set of samples $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ drawn from $\mathcal{M}$, we say $\mathcal{T}$* perfectly clusters *the samples if there are no two samples $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ such that both $\boldsymbol{x}_i$ and $\boldsymbol{x}_{i'}$ are drawn from the same component but associated to different leaf nodes.*

6.3. Analysis of RECURSIVESPECTRALPROJECTION

At a high level, the clustering algorithm proceeds as follows. First, the algorithm draws k sets of N i.i.d. samples, where

$$N = \widetilde{O}\left(k^4(d + \log k/\delta)\max\left(\frac{1}{\epsilon^4}, \frac{1}{\zeta^2}\right)\right) ,$$

function RECURSIVESPECTRALPROJECTION($S^1, \dots, S^k, \delta$)
  Let $\mathcal{T}_0$ be the empty tree.
  for $\ell = 1, \dots, k$ **do**
    Cluster samples in S^ℓ using $\mathcal{T}_{\ell-1}$ into clusters $C_1^\ell, \dots, C_{k''}^\ell$ for some $k'' \le k$.
    Each cluster is associated with a leaf node of $\mathcal{T}_{\ell-1}$.
    for $m = 1, \dots, k''$ **do**
      $\widehat{\boldsymbol{\mu}}^\ell(C_m^\ell) \leftarrow \frac{1}{|C_m^\ell|} \sum_{i \in C_m^\ell} \boldsymbol{x}_i$
      Let $\boldsymbol{B}_m^\ell$ be the matrix whose columns are $\frac{1}{\sqrt{|C_m^\ell|}}(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}^\ell(C_m^\ell))$ for $i \in C_m^\ell$.
      $\{\boldsymbol{v}\} = \text{APPROXPCA}(\boldsymbol{B}_m^\ell, 1, 1/10)$.
      Let $\lambda = \boldsymbol{v}^T \left(\boldsymbol{B}_m^\ell\right) \left(\boldsymbol{B}_m^\ell\right)^T \boldsymbol{v}$.
      if $\lambda \le O(k^3 \log(kn/\delta))$ **then**
        ▷ In this case we return a single cluster
        Do not split the samples
      else
        ▷ In this case, we partition the samples and recursively cluster each partition.
        Project the $\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}^\ell(C_m^\ell)$ onto $\boldsymbol{v}$ and sort them.
        Find the maximum gap between two consecutive values $\boldsymbol{v}^T(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}^\ell(C_m^\ell))$.
        Associate to leaf node m the function $\text{label}(v) = v^T(\boldsymbol{x} - \widehat{\boldsymbol{\mu}}^\ell(C_m^\ell))$
        Attach two new child nodes to node m in $\mathcal{T}_\ell$.
    Let $\mathcal{T}_\ell$ be the new tree.
    if $\mathcal{T}_\ell = \mathcal{T}_{\ell-1}$ **then**
      return $\mathcal{T}_\ell$ and the samples S^ℓ.

and divides them into sets $S_1, \dots, S_k$. The algorithm then iteratively builds a clustering tree using these sets of samples one at a time, never reusing a previous set of samples. In a nutshell, the resulting tree is a valid clustering tree for the underlying mixture with error probability $O(\text{poly}(\delta, 1/k))$, so that only components with means that are suitably close together land in the same cluster. However, we do not wish to put any constraints on the weights or means of the true, underlying GMM. In this regime it is not straightforward to prove a statement directly in terms of the underlying parameters.[10] Therefore, our result must be stated in a way that depends on the samples drawn, but we show that this is sufficient for our purposes.

Since our results are stated relative to the specific set of samples drawn, it will be useful to define the following quantities. For any $\ell = 1, \dots, k$, let $\widehat{w}_j^\ell$ be the fraction of samples in the set S_ℓ that come from component j. For any set of components $C \subseteq [k]$, let $\widehat{\boldsymbol{\mu}}^\ell(C)$ be the empirical mean of the samples from set S_ℓ that come from C, and let $\widehat{w}^\ell(C) = \sum_{j \in C} \widehat{w}_j^\ell$. Let

$$\widetilde{\boldsymbol{\mu}}^\ell(C) \;=\; \sum_{j \in C} \frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C)} \boldsymbol{\mu}_j$$

be the mean within cluster C using the empirical weights instead of the true weights.

10. Consider a mixture component with very small weight but very large mean. It does not contribute meaningfully to the density of the overall mixture but significantly affects terms involving all underlying means and weights.

We also define

$$\widetilde{\boldsymbol{C}}^{\ell}(C) \;=\; \sigma^2 \boldsymbol{I} + \sum_{j \in C} \frac{\widehat{w}_j^{\ell}}{\widehat{w}^{\ell}(C)} \left(\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^{\ell}(C)\right)\left(\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^{\ell}(C)\right)^T \;, \text{ and}$$

$$\gamma^{\ell}(C) = \max_{j \in C} \frac{\widehat{w}_j^{\ell}}{\widehat{w}^{\ell}(C)} \left\|\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^{\ell}(C)\right\|_2^2 \;.$$

With this, we may now state the key result for this section:

Theorem 54 *Fix $\delta > 0$. Let $S_1, \ldots, S_k$ be independent sets of samples, where each S_i contains n i.i.d. samples from $\mathcal{M}$, and where n is chosen sufficiently large so that*

$$n \geq c\left(\frac{d + \log(1/\delta) + \log(kn)}{\epsilon^2}\right) \;,$$

for some universal constant c. Let $\mathcal{T} = \text{RECURSIVESPECTRALPROJECTION}(S_1, \ldots, S_k, \delta)$, and let t be the iteration in which the algorithm terminates. Then, with probability $1 - \delta$, we have that $\mathcal{T}$ perfectly clusters S_t. Moreover, if $C_1, \ldots, C_{k'}$ is the clustering induced by $\mathcal{T}$, we have that $\gamma_t(C_p) \leq O(k^3 \log(k/\delta))$, for all $p = 1, \ldots, k'$ satisfying $\widehat{w}_t(C_p) \geq O(\epsilon/k)$.

Our analysis consists of two parts. First, we show that if we extend the current clustering tree in a given iteration, then the extended tree remains a clustering tree for $\mathcal{M}$ with error probability $1 - \delta$. Next, we show that if we terminate in iteration t, we also have $\gamma_t \leq O(k^3 \log(k/\delta))$.

Proof [Proof of Theorem 54] We proceed inductively to show that the following property holds in step $\ell \in [k]$ with probability at least $1 - O(\delta/k)$: If $\mathcal{T}_{\ell-1}$ is a clustering tree with error $O(\delta/\text{poly}(k, N))$ and associated clustering $C_1, \ldots, C_m$, then:

1. For every cluster C_p with $\widehat{w}^{\ell}(C_p) \geq O(\epsilon/k)$ and $\gamma^{\ell}(C_p) \geq O(k^2 \log(k/\delta'))$, we output a clustering direction with error $\delta/(kN)$.

2. For every cluster C_p with $\widehat{w}^{\ell}(C_p) \geq O(\epsilon/k)$ for which we do not output a clustering direction, we must have $\gamma^{\ell}(C_p) \leq O(k^3 \log(k/\delta'))$.

First, since $\mathcal{T}_{\ell-1}$ is a clustering tree with error $O(\delta/\text{poly}(N, k))$, we may assume with probability $1 - \delta/k$ that it misclassifies no points in S_{ℓ}. Fix any cluster C_p. We may assume that some nonzero number of samples from the set S_{ℓ} land in the cluster C_p, as otherwise the algorithm trivially satisfies the desired guarantees in this iteration for this cluster. Now, restricted to the components in the cluster C_p, the resulting distribution is still a GMM with pdf given by

$$\mathcal{M}_{C_p} = \sum_{j \in C_p} \frac{w_j}{w(C_p)} \mathcal{N}_{\boldsymbol{\mu}_j, \tau_j}(\boldsymbol{x}) \;.$$

Moreover, in the set S_{ℓ} we have $\widehat{w}^{\ell}(C_p) \cdot n$ samples from this GMM $\mathcal{M}_{C_p}$. Let $S_{\ell,p}$ denote this set of samples. Moreover, define

$$\widehat{\boldsymbol{C}}_p^{\ell} = \frac{1}{\widehat{w}^{\ell}(C) n} \sum_{i \in S_{\ell,p}} (\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}^{\ell}(C))(\boldsymbol{x}_i - \widehat{\boldsymbol{\mu}}^{\ell}(C))^T$$

to be the empirical covariance within this cluster C_p. Lemma 45 implies that with probability $1 - O(\delta/k)$, we have

$$\begin{aligned}\left\|\widehat{\boldsymbol{C}}_p^\ell - \widetilde{\boldsymbol{C}}^\ell(C_p)\right\|_{\mathrm{op}} &\le O\left(k^{1/2}\frac{\epsilon}{\sqrt{\widehat{w}^\ell(C_p)}} + \frac{\epsilon}{\sqrt{\widehat{w}^\ell(C_p)}}k\gamma^\ell(C_p)^{1/2} + k\frac{\epsilon^2}{\widehat{w}^\ell(C_p)}\right)\\ &\le O\left(k\sqrt{\epsilon} + k\gamma^\ell(C_p)^{1/2}\sqrt{\epsilon}\right)\end{aligned}$$

for all C_p with $\widehat{w}^\ell(C_p) \ge \epsilon/k$, since we are assuming throughout this entire section that $\epsilon < 1/k$.

We now show that this concentration suffices to give a clustering direction if the components within this cluster are too far apart:

Lemma 55 *Let $\beta \ge \sigma k^2 \log(kn/\delta)$.*

Assume that $\gamma^\ell(C_p) \ge 5\sigma\beta = \Theta(\beta)$, and let $\boldsymbol{A}$ be a matrix such that $\|\boldsymbol{A} - \widetilde{\boldsymbol{C}}^\ell(C_p)\|_{\mathrm{op}} \le O\left(\beta\sqrt{\epsilon} + k\gamma^\ell(C_p)^{1/2}\sqrt{\epsilon}\right)$. Then $\|\boldsymbol{A}\|_{\mathrm{op}} \ge \Omega(\beta)$. Moreover, if $\boldsymbol{v}$ is any direction so that $\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v} \ge 4\beta = \Theta(\beta)$, then it is a clustering direction for $\mathcal{M}_{C_p}$ with error probability δ/kn.

Proof We start with the first claim. Recall that $\gamma^\ell(C_p) = \max_j \frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}\|\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\|_2^2$. Let i be the corresponding index, i.e.,

$$i = \arg\max_{j\in[k]} \sqrt{\frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}}\|\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\|_2^2 \,,$$

and let $\boldsymbol{u} = \frac{\boldsymbol{\mu}_i - \widetilde{\boldsymbol{\mu}}^\ell(C_p)}{\|\boldsymbol{\mu}_i - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\|_2}$. Then

$$\begin{aligned}\boldsymbol{u}^T\boldsymbol{A}\boldsymbol{u} &\ge \boldsymbol{u}^T\widetilde{\boldsymbol{C}}^\ell(C_p)\boldsymbol{u} - O\left(\beta\sqrt{\epsilon} + k\gamma^\ell(C_p)^{1/2}\sqrt{\epsilon}\right)\\ &= \sum_{j\in C_p}\frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}\langle \boldsymbol{u}, \boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\rangle^2 + 1 - O\left(\beta\sqrt{\epsilon} + k\gamma^\ell(C_p)^{1/2}\sqrt{\epsilon}\right)\\ &\ge \gamma^\ell(C_p)/\sigma + 1 - O\left(\beta\sqrt{\epsilon} + k\gamma^\ell(C_p)^{1/2}\sqrt{\epsilon}\right)\\ &\ge 4\beta \,.\end{aligned}$$

This establishes that $\|\boldsymbol{A}\|_{\mathrm{op}} \ge 4\beta$.

Next, let $\boldsymbol{v}$ be a direction satisfying $\boldsymbol{v}^T\boldsymbol{A}\boldsymbol{v} \ge 4\sigma\beta$. Then by a similar argument as above, we know that

$$\sum_{j\in C_p}\frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}\langle \boldsymbol{v}, \boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\rangle^2 \ge 2\beta \,.$$

We now claim that there must be indices i and i' so that

$$|\langle \boldsymbol{v}, \boldsymbol{\mu}_i\rangle - \langle \boldsymbol{v}, \boldsymbol{\mu}_{i'}\rangle| \ge \sqrt{\beta} = O(k\sqrt{\log(nk/\delta)}) \,.$$

By basic Gaussian concentration (see Fact 39), the maximum of n samples from a univariate Gaussian with variance σ deviates by at most $\sigma\sqrt{\log(nk/\delta)}$ from the mean with probability $1 - \delta'/kn$. Hence v is a clustering direction for $\mathcal{M}_{C_p}$ with error probability δ/kn as stated in the lemma.

Suppose that the claim is not true. Since $\widetilde{\boldsymbol{\mu}}^\ell(C_p)$ is a weighted average of the $\boldsymbol{\mu}_j$, this implies that

$$|\langle \boldsymbol{v}, \boldsymbol{\mu}_i\rangle - \langle \boldsymbol{v}, \widetilde{\boldsymbol{\mu}}^\ell(C_p)\rangle| \;\le\; \sqrt{\beta}\,.$$

But then we have

$$\sum_{j\in C_p} \frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}\langle \boldsymbol{v}, \boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p)\rangle^2 \;\le\; \sum_{j\in C_p} \frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}\beta \;=\; \beta\,,$$

which is a contradiction. ∎

The lemma shows that if $\gamma^\ell(C_p)$ is sufficiently large, and we have an estimate $\boldsymbol{A}$ of the matrix $\widetilde{\boldsymbol{C}}^\ell(C_p)$ up to spectral error $O(\beta)$, then $\|\boldsymbol{A}\|_{\mathrm{op}}$ is large, and our algorithm produces a clustering direction for $\mathcal{M}_{C_p}$. We now establish the converse: if the spectral norm of $\boldsymbol{A}$ is large, then $\gamma_\ell(C_p)$ must be large. We show this implication by proving the contrapositive:

Lemma 56 *Let $\boldsymbol{A}$ be so that $\|\boldsymbol{A} - \widetilde{\boldsymbol{C}}^\ell(C_p)\|_{\mathrm{op}} \le O(\beta)$. Then $\|\boldsymbol{A}\|_{\mathrm{op}} \le O(k\gamma^\ell(C_p) + \beta + 1)$.*

Proof Indeed, we have

$$\begin{aligned}
\|\boldsymbol{A}\|_{\mathrm{op}} \;&\le\; \|\widetilde{\boldsymbol{C}}^\ell(C_p)\|_{\mathrm{op}} + O(\beta)\\
&\le \left\|\sum_{j\in C_p} \frac{\widehat{w}_j^\ell}{\widehat{w}^\ell(C_p)}(\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p))(\boldsymbol{\mu}_j - \widetilde{\boldsymbol{\mu}}^\ell(C_p))^T\right\|_{\mathrm{op}} + 1 + O(\beta)\\
&\le \sum_{j\in C_p} \gamma^\ell(C_p) + 1 + O(\beta)\\
&\le\; O(k\gamma^\ell(C_p) + \beta)
\end{aligned}$$

as claimed. ∎

Putting this together, we have:

Corollary 57 *Let $\boldsymbol{B} \in \mathbb{R}^{d\times n}$ be a matrix and let $\boldsymbol{A} = \boldsymbol{B}\boldsymbol{B}^T$. Assume we have*

$$\|\boldsymbol{A} - \widetilde{\boldsymbol{C}}^\ell(C_p)\|_{\mathrm{op}} \;\le\; O(k\log(k/\delta)\sqrt{\epsilon} + \left(\gamma^\ell(C_p)\right)^{1/2}\sqrt{\epsilon})\,.$$

Then one of the following two cases holds: (i) If we also have $\|\boldsymbol{A}\|_{\mathrm{op}} \le O(k\beta)$, then $\gamma_\ell(C_p) \le O(k\beta) = O(k^3\log(kn/\delta))$. (ii) Otherwise, ApproxPCA$(\boldsymbol{B}, 1, 1/10)$ *produces a clustering direction with error probability δ/kn.*

Proof The first conclusion follows from Lemma 55. On the other hand, suppose that $\|\boldsymbol{A}\|_{\mathrm{op}} \ge \Omega(k\beta)$. Then by Lemma 55, we have $\gamma_\ell(C_p) \ge \Omega(\beta)$ which by Lemma 55 gives the desired conclusion. ∎

This proves the induction. Note that this implies that RecursiveSpectralProjection can only add at most k leaves, since each time it must peel off at least one component. Repeated application of this result, and union bounding over all recursive calls of RecursiveSpectralProjection finally completes the proof of Theorem 54. ∎

7. Finding the subspace spanned by the means

In this section, let t denote the iteration in which RECURSIVESPECTRALPROJECTION terminates. Let $C_1, \ldots, C_{k'}$ be the final clustering returned by RECURSIVESPECTRALPROJECTION, and let the set of samples from set S_t belonging to cluster C_p be denoted $S_{t,p}$. By the above, we may assume that our algorithm knows this clustering perfectly. Moreover, for each p with $\widehat{w}^t(C_p) \geq O(\epsilon/k)$, we have $\gamma^t(C_p) = O(k^3 \log(k/\delta))$. It will be useful for us to reuse the samples in S_t because we then know that the $\widehat{w}_t$ quantities are the same as the criteria used in the previous clustering step.

We now give an algorithm to find the subspace spanned by the means of the Gaussian mixture within any cluster C_p satisfying $\gamma_t(C_p) \geq O(\epsilon/k)$. Moreover, the algorithm also gives a much finer estimate for σ.

We denote subspaces with upper case letters. For a subspace V, we let $V^\perp$ be the orthogonal complement of V. We define Π_V to be an orthogonal projection onto the subspace V.

Algorithm 4 Algorithm for approximating finding the subspace spanned by the means

function FINDAPPROXSUBSPACEANDCOVARIANCE($k, S, \epsilon, \delta, \zeta$)
  $\widehat{\boldsymbol{\mu}} = \frac{1}{|S|} \sum_{i \in S} \boldsymbol{x}_i$.
  Let $\boldsymbol{A}$ be the matrix whose columns are $\frac{\boldsymbol{x}_j - \widehat{\boldsymbol{\mu}}}{\sqrt{|S|}}$.
  $V \leftarrow$ APPROXPCA$(\boldsymbol{A}, k+1, \min(\epsilon^2/k^2, \zeta/k))$.
  Let $\boldsymbol{v}_{k+1}$ be the $(k+1)$-st vector returned. Let $\widehat{\sigma}^2 \leftarrow \boldsymbol{v}_{k+1}^T \boldsymbol{A}\boldsymbol{A}^T \boldsymbol{v}_{k+1}$.
  return Π_V, an orthogonal projector onto the subspace V, and $\widehat{\sigma}^2$

Lemma 58 *Fix $\epsilon, \delta > 0$. Let p be so that $\widehat{w}^t(C_p) \geq O(\epsilon/k)$. Then, with probability $1 - \delta$, if V is the subspace output by* FINDAPPROXSUBSPACEANDCOVARIANCE$(k, S_{t,p}, \epsilon, \delta, \zeta)$, *we have*

$$\|\Pi_{V^\perp}(\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}}(C_p))\|_2 \;\leq\; O\left(\frac{\epsilon}{k^{1/2}\sqrt{\widehat{w}_j^t}}\right) \qquad \text{for all } j \in C_p\,. \tag{13}$$

Moreover, if $\widehat{w}^t(C_p) \geq \Omega(1/k)$, we have that $|\widehat{\sigma} - \sigma| \leq (1+\zeta)\sigma$.

Proof By Lemma 45, since we assume that $\gamma^t(C_p) \leq \widetilde{O}(k^3)$ we know that we have except with probability $1 - \delta$,

$$\|\boldsymbol{A}\boldsymbol{A}^T - \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}} \;\leq\; \frac{1}{\widehat{w}^t(C_p)} \min\left(\frac{\epsilon^2}{k^2}, \frac{\zeta}{k}\right)\,.$$

The second claim then immediately follows from the guarantees of APPROXPCA. By the guarantees of APPROXPCA, we know that

$$\|\boldsymbol{\Pi}_U \boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{A}^T\|_{\mathrm{op}} \;\leq\; \left(1 + \frac{1}{\widehat{w}^t(C_p)}\epsilon^2/k\right)\sigma_{k+1}(\boldsymbol{A})$$

where $\sigma_{k+1}(\boldsymbol{A})$ denotes the $(k+1)$-st largest singular value of $\boldsymbol{A}$. Because

$$\|\boldsymbol{A}\boldsymbol{A}^T - \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}} \;\leq\; \frac{1}{\widehat{w}^t(C_p)}\epsilon^2/k$$

this immediately implies (since the $k+1$st eigenvalue of $\boldsymbol{C}$ is 1) that

$$\begin{aligned}\sigma_{k+1}(\boldsymbol{A}) &\leq \sigma^2\left(1+\frac{1}{\widehat{w}^t(C_p)}\epsilon^2/k\right)^2 \\ &= \sigma^2\left(1+\frac{1}{\widehat{w}^t(C_p)}O(\epsilon^2/k)\right),\end{aligned}$$

where the last line follows since $\frac{1}{\widehat{w}^t(C_p)}\epsilon^2/k = O(\epsilon)$ by our assumption about $\widehat{w}^t(C_p)$, and since $(1+x)^2 = (1+O(x))$ for $x \geq 0$ small.

Hence since $\|\boldsymbol{\Pi}_U \boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{\Pi}_U \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}} \leq \|\boldsymbol{A}\boldsymbol{A}^T - \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}}$, we have that by the triangle inequality,

$$\begin{aligned}\|\boldsymbol{\Pi}_U \widetilde{\boldsymbol{C}}^t(C_p) - \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}} &\leq \|\boldsymbol{\Pi}_U \widetilde{\boldsymbol{C}}^t(C_p) - \boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{A}^T\|_{\mathrm{op}} + \|\boldsymbol{\Pi}\boldsymbol{A}\boldsymbol{A}^T - \boldsymbol{A}\boldsymbol{A}^T\|_{\mathrm{op}} + \|\boldsymbol{A}\boldsymbol{A}^T - \widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}} \\ &\leq \sigma^2(1+\frac{1}{\widehat{w}^t(C_p)}O(\epsilon^2/k)) .\end{aligned}$$

Equivalently, this gives

$$\|\boldsymbol{\Pi}_{U^\perp}\widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}}{}^2 \leq \sigma^4(1+\frac{1}{\widehat{w}^t(C_p)}O(\epsilon^2/k))^2 = \sigma^4(1+\frac{1}{\widehat{w}^t(C_p)}O(\epsilon^2/k)) .$$

By definition, we have

$$\begin{aligned}\|\boldsymbol{\Pi}_{U^\perp}\widetilde{\boldsymbol{C}}^t(C_p)\|_{\mathrm{op}}{}^2 &= \max_{\|\boldsymbol{v}\|_2=1} \boldsymbol{v}^T\boldsymbol{\Pi}_{U^\perp}\widetilde{\boldsymbol{C}}^t(C_p)\widetilde{\boldsymbol{C}}^t(C_p)^T\boldsymbol{\Pi}_{U^\perp}^T\boldsymbol{v} \\ &= \max_{\|\boldsymbol{v}\|_2=1} \boldsymbol{v}^T\boldsymbol{\Pi}_{U^\perp}(\boldsymbol{M}+\sigma^2\boldsymbol{I})(\boldsymbol{M}+\sigma^2\boldsymbol{I})^T\boldsymbol{\Pi}_{U^\perp}^T\boldsymbol{v} \\ &= \max_{\|\boldsymbol{v}\|_2=1} (\boldsymbol{v}^T\boldsymbol{\Pi}_{U^\perp}\boldsymbol{M}^2\boldsymbol{\Pi}_{U^\perp}^T\boldsymbol{v} + 2\sigma^2\boldsymbol{v}^T\boldsymbol{\Pi}_{U^\perp}\boldsymbol{M}\boldsymbol{\Pi}_{U^\perp}^T\boldsymbol{v} + \sigma^4\boldsymbol{v}^T\boldsymbol{\Pi}_{U^\perp}\boldsymbol{\Pi}_{U^\perp}^T\boldsymbol{v}) \\ &= \max_{\|\boldsymbol{u}\|_2=1,\boldsymbol{u}\in U} \boldsymbol{u}^T\boldsymbol{M}^2\boldsymbol{u} + 2\sigma^2\boldsymbol{u}^T\boldsymbol{M}\boldsymbol{u} + \sigma^4 ,\end{aligned}$$

where $\boldsymbol{M} = \sum_{j=1}^k \frac{\widehat{w}_j^t}{\widehat{w}^t(C_p)}(\boldsymbol{\mu}_j-\boldsymbol{\mu}')(\boldsymbol{\mu}_j-\boldsymbol{\mu}')^T$. Since this quantity is at most $(1+\frac{1}{\widehat{w}^t(C_p)}O(\epsilon^2/k))\sigma^4$ and since $\sigma = \Theta(1)$, this implies that

$$\max_{\|\boldsymbol{u}\|_2=1,\boldsymbol{u}\in U} \boldsymbol{u}^T\boldsymbol{M}\boldsymbol{u} \leq \frac{1}{\widehat{w}_t(C_p)}O(\epsilon^2/k)$$

which is equivalent to the fact that $\|\boldsymbol{\Pi}_{U^\perp}\boldsymbol{M}\boldsymbol{\Pi}_{U^\perp}^T\|_{\mathrm{op}} \leq \frac{1}{\widehat{w}_t(C_p)}O(\epsilon^2/k)$. For all j, we have $\widehat{w}_j(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j-\boldsymbol{\mu})^T \preceq M$, and so

$$\frac{\widehat{w}_j^t}{\widehat{w}^t(C_p)}\|\boldsymbol{\Pi}_{U^\perp}(\boldsymbol{\mu}_j-\boldsymbol{\mu})\|_2^2 \leq \|\boldsymbol{\Pi}_{U^\perp}\boldsymbol{M}\boldsymbol{\Pi}_{U^\perp}^T\|_{\mathrm{op}} \leq \frac{1}{\widehat{w}^{k'}(C_p)}O(\epsilon^2/k)$$

and thus

$$\|\boldsymbol{\Pi}_{S^\perp}(\boldsymbol{\mu}_j-\boldsymbol{\mu}')\|_2 \leq O\left(\frac{\epsilon}{k^{1/2}\left(\widehat{w}_j^t\right)^{1/2}}\right) .$$

We can then apply Corollary 44 (with parameter ϵ^2/k) and another triangle inequality to conclude the proof. ■

We now show that this implies that projecting onto this subspace U can only affect the total variation distance by at most $O(\epsilon)$.

Theorem 59 *Let $\mathcal{M}_{\theta^*}$ denote the underlying mixture, and let U be a subspace which satisfies (13) after $O(k^4(d+\log(k/\delta))/\epsilon^4)$ samples. Then with probability $1-\delta$, we have $\|\mathcal{M}_{\theta_1}-\mathcal{M}_{\theta_S}\|_1 \le O(\epsilon)$ where θ_1 has the same parameters θ^* except the mean of component j is $\boldsymbol{\mu}_j-\widehat{\mu}$, and θ_U is the mixture with the same parameters as θ^*, except for all j, component j has mean $\boldsymbol{\Pi}_S(\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}})$.*

Proof Let θ_1' be equal to θ_1 except if component j had $w_j \le O(\epsilon/k)$ we now set $w_j = 0$, and let θ_S' be the same for θ_U. Let J be the set of components which have nonzero weight after this step. [11] By the triangle inequality, it is clear that it suffices to show that $\|\mathcal{M}_{\theta_1'} - \mathcal{M}_{\theta_S'}\|_1 \le O(\epsilon)$. By Bernstein's inequality, since we take $O(k^4(d+\log(k/\delta))/\epsilon^4)$ samples, we know that for all j, with probability $1-\delta/k$, we have that $|\widehat{w}_j^t - w_j| \le O(\epsilon/k)$ and thus by a union bound, we know that $\widehat{w}_j^t \ge \Omega(w_j)$ for all j. Thus in particular by Lemma 58, we know that for all j with $w_j \neq 0$,

$$\begin{aligned}\|\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}} - \boldsymbol{\Pi}_U(\boldsymbol{\mu}_j - \widehat{\boldsymbol{\mu}})\|_2 &\le O\left(\frac{\epsilon}{k^{1/2}\sqrt{\widehat{w}_j^t}}\right) \\ &\le O\left(\frac{\epsilon}{k^{1/2}w_j^{1/2}}\right) .\end{aligned}$$

We now have

$$\begin{aligned}\|\mathcal{M}_{\theta_1} - \mathcal{M}_{\theta_S}\|_1 &\le \|\mathcal{M}_{\theta_1'} - \mathcal{M}_{\theta_S'}\|_1 + 2\epsilon \\ &= \left\|\sum_{j\in J} w_j\left(\mathcal{N}_{\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}},\sigma} - \mathcal{N}_{\boldsymbol{\Pi}_U(\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}),\sigma}\right)\right\|_1 + 2\epsilon \\ &\le \sum_{j\in J} w_j \left\|\mathcal{N}_{\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}},\sigma} - \mathcal{N}_{\boldsymbol{\Pi}_U(\boldsymbol{\mu}_j-\widehat{\boldsymbol{\mu}}),\sigma}\right\|_1 + 2\epsilon \\ &\overset{(a)}{\le} \sum_{j\in J} w_j O\left(\frac{\epsilon}{k^{1/2}w_j^{1/2}}\right) + 2\epsilon \\ &\le \sum_{j\in J} O\left(\frac{w_j^{1/2}\epsilon}{k^{1/2}}\right) + 2\epsilon \\ &\le O(\epsilon) ,\end{aligned}$$

where (a) follows by Fact 35. ■

11. This is not technically a valid parameter set because the w_i do not sum to 1, but the meaning should be clear, we simply work with the associated subdistributions.

Thus we have shown that we can find this subspace and project onto it to essentially reduce the problem to a k-dimensional problem. We also make a further simplification: because we now have an estimate of σ up to multiplicative $(1+\zeta)$ error, it can be shown that this error is negligible, and so for simplicity of exposition we will assume for the rest of the section that we have σ exactly. Thus we will assume for the rest of the section that $\sigma = 1$.

8. The k-dimensional system of polynomial inequalities

We now give an outline of the subroutine FITPOLYPROGRAMMULTIVARIATE. In a nutshell, we extend our system of polynomial inequalities from Section 3 to the k-dimensional setting. To decouple this k-dimensional analysis from the surrounding d-dimensional algorithm, we assume that we are drawing samples from a k-dimensional k-GMM with parameters $\theta^\dagger$.

As before, our system of polynomial inequalities is based on univariate density estimates. First, we show that we can simultaneously estimate a large number of directions with only a modest overhead in the sample complexity. In the following sequence of lemmas, we let $0 < C_1 < C_2 < C_3 < 1$ be constants that we do not specify further.

Lemma 60 *Let $V \subset S^{k-1}$ be a fixed set of directions. Moreover, let $\epsilon > 0$ and let $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n$ be samples from the distribution $\mathcal{M}_{\theta^\dagger}$ where*

$$n = \widetilde{O}\left(\frac{k + \log(|V|/\delta)}{\zeta^2}\right).$$

For all $\boldsymbol{v} \in V$, let $p_{\mathrm{dens},v}$ be the result of ESTIMATEDENSITY(k, ζ, δ') *on the samples $\langle \boldsymbol{x}_1, v\rangle, \dots, \langle \boldsymbol{x}_n, v\rangle$ for $\delta' = \delta/|V|$. Then with probability $1-\delta$, we have*

$$\|p_{\mathrm{dens},v} - \mathcal{M}_{\boldsymbol{v}\cdot\theta^\dagger}\|_1 \;\le\; C_1 \cdot \zeta$$

for all $v \in V$.

Proof The guarantee of ESTIMATEDENSITY for any fixed direction $\boldsymbol{v} \in V$ immediately gives the desired L_1-approximation guarantee with probability $1-\delta'$. The claim then follows from a union bound over all directions $\boldsymbol{v} \in V$. ■

Using these univariate density estimate $p_{\mathrm{dens},v}$, we now consider the following system of polynomial inequalities. This system $S_{k\text{-dim},V}$ is essentially a conjunction of the univariate constraints for each direction. For simplicity, we only state the version for well-behaved parameters here. We adopt the same notation as in Section 3.

$$\begin{aligned} S_{k\text{-dim},V}(\nu) \;=\; S_{V,K,p,\mathcal{P},S}(\nu) = \forall a_1^{(1)}, \dots a_K^{(1)}, b_1^{(1)}, \dots, b_K^{(1)}, \;\dots,\; a_1^{(|V|)}, \dots a_K^{(|V|)}, b_1^{(|V|)}, \dots, b_K^{(|V|)} \;:\\ \exists d_1^{(1)}, \dots, d_s^{(1)}, \xi_1^{(1)} \dots \xi_t^{(1)} \;\dots\; d_1^{(|V|)}, \dots, d_s^{(|V|)}, \xi_1^{(|V|)} \dots \xi_t^{(|V|)} \;:\\ \text{valid-parameters}_S(\theta)\\ \wedge \bigwedge_{i\in[|V|]} \text{correct-breakpoints}_{\mathcal{P}}(\theta, d^{(i)})\\ \wedge \bigwedge_{i\in[|V|]} \mathcal{A}_K\text{-bounded}_{p,\mathcal{P}}(\theta, \nu, a^{(i)}, b^{(i)}, c^{(i)}, d^{(i)}, \xi^{(i)}) \,. \end{aligned}$$

By construction of $S_{k\text{-dim},V}$, there is a set of k-dimensional k-GMM parameters such that the constraints are satisfied for sufficiently large ν. In particular, the parameters $\theta^\dagger$ satisfy $S_{k\text{-dim},V}(\nu)$ for $\nu = O(\zeta)$. The proof of the following lemma follows the same argument as in the univariate case, so we omit further details.

Lemma 61 *There is a set of k-GMM parameters $\theta \in \Theta_k$ such that all constraints in $S_{k\text{-dim},V}(\nu)$ are satisified for $\nu = C_2 \cdot \zeta$.*

Next, we show that a set of GMM parameters satisfying $S_{k\text{-dim},V}$ is close to $\mathcal{M}_{\theta^\dagger}$.

Lemma 62 *Let $\theta \in \Theta_k$ be a set of k-GMM parameters such that all constraints in $S_{k\text{-dim},V}(\nu)$ are satisified for $\nu \leq C_3 \cdot \zeta$. Then we have $\|\mathcal{M}_\theta - \mathcal{M}_{\theta^\dagger}\|_1 \leq \epsilon$.*

Proof Since θ is feasible for $S_{k\text{-dim},V}(C_3 \cdot \zeta)$, we have $\|\mathcal{M}_{v\cdot\theta} - \mathcal{M}_{v\cdot\theta^\dagger}\|_1 \leq \frac{\zeta}{2} < \zeta$ for all directions $v \in V$ by the construction of $S_{k\text{-dim},V}$.

Now assume $\|\mathcal{M}_\theta - \mathcal{M}_{\theta^\dagger}\|_1 \geq \epsilon$. Since ζ is defined as

$$\zeta(\epsilon, k, V) \;=\; \inf_{\theta', \theta'' : \|\mathcal{M}_{\theta'} - \mathcal{M}_{\theta''}\|_1 \geq \epsilon} \max_{\boldsymbol{v} \in V} \|\mathcal{M}_{\boldsymbol{v}\cdot\theta'} - \mathcal{M}_{\boldsymbol{v}\cdot\theta''}\|_1$$

we get

$$\zeta(\epsilon, k, V) \;\leq\; \max_{\boldsymbol{v} \in V} \|\mathcal{M}_{\boldsymbol{v}\cdot\theta} - \mathcal{M}_{\boldsymbol{v}\cdot\theta^\dagger}\|_1 \;<\; \zeta,$$

which gives a contradiction. Hence $\|\mathcal{M}_\theta - \mathcal{M}_{\theta^\dagger}\|_1 < \epsilon$. ■

We now translate this approximation guarantee in the k-dimensional space of $\mathcal{M}_{\theta^\dagger}$ back to the original d-dimensional space. We use the following notation: for a matrix $\mathbf{\Pi} \in \mathbb{R}^{k \times d}$, we write $\mathbf{\Pi}\theta$ for the set of k-GMM parameters where the matrix $\mathbf{\Pi}$ is applied to each component mean.

Theorem 63 *Let $\widehat{\theta} \in \Theta_k$ be the set of k-GMM parameters returned by* FITPOLYPROGRAMMULTIVARIATE. *Moreover, let $\mathbf{\Pi} \in \mathbb{R}^{k \times d}$ be an orthogonal projection matrix such that $\theta^\dagger = \mathbf{\Pi}\theta^*$.Then*

$$\|\mathcal{M}_{\mathbf{\Pi}^T\mathbf{\Pi}\theta^*} - \mathcal{M}_{\mathbf{\Pi}^T\widehat{\theta}}\|_1 \;\leq\; O(\epsilon)\;.$$

Proof To simplify notation, we write $\mathcal{N}_{\Pi,\boldsymbol{\mu}}(x)$ for the multivariate Gaussian pdf with covariance matrix $\boldsymbol{I}$ and mean $\mathbf{\Pi}\boldsymbol{\mu}$, i.e., the pdf of the Gaussian distribution in the space spanned by the projector $\mathbf{\Pi}$. Similarly, we write $\mathcal{N}_{\overline{\mathbf{\Pi}},\boldsymbol{\mu}}(x)$ for the pdf of the Gaussian distribution in the orthogonal subspace. Since we assume a spherical covariance matrix for each component, we have

$$\mathcal{N}_{\boldsymbol{\mu}}(\boldsymbol{x}) \;=\; \mathcal{N}_{\boldsymbol{I},\boldsymbol{\mu}}(\boldsymbol{x}) \;=\; \mathcal{N}_{\mathbf{\Pi},\boldsymbol{\mu}}(\boldsymbol{x}) \cdot \mathcal{N}_{\overline{\mathbf{\Pi}},\boldsymbol{\mu}}(\boldsymbol{x})\;,$$

which follows directly from the definition of the multivariate normal pdf and Pythagoras' Theorem.

We now use this identity to factorize our integrand in the L_1-difference between $\mathcal{M}_{\mathbf{\Pi}^T\mathbf{\Pi}\theta^*}$ and $\mathcal{M}_{\mathbf{\Pi}^T\widehat{\theta}}$:

$$\begin{aligned}
&\int_{\boldsymbol{x}} |\mathcal{M}_{\mathbf{\Pi}^T\mathbf{\Pi}\theta^*}(\boldsymbol{x}) - \mathcal{M}_{\mathbf{\Pi}^T\widehat{\theta}}(\boldsymbol{x})| \,\mathrm{d}\boldsymbol{x} \\
&\quad= \int_{\boldsymbol{x}} \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(x) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \right| \mathrm{d}\boldsymbol{x} \\
&\quad= \int_{\boldsymbol{x}} \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}) \cdot \mathcal{N}_{\overline{\mathbf{\Pi}},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}^*}(\boldsymbol{x}) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \cdot \mathcal{N}_{\overline{\mathbf{\Pi}},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \right| \mathrm{d}\boldsymbol{x} \\
&\quad= \int_{\boldsymbol{x}} \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}) \cdot \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \cdot \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}) \right| \mathrm{d}\boldsymbol{x} \\
&\quad= \int_{\boldsymbol{x}} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}) \cdot \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \right| \mathrm{d}\boldsymbol{x} \,.
\end{aligned}$$

Next, we decompose the d-dimensional integral into two parts. One part (using integration variable $\boldsymbol{x}_2$) is over the space spanned by the projector $\mathbf{\Pi}$, the other part (using integration variable $\boldsymbol{x}_1$) is over the orthogonal complement of $\mathbf{\Pi}$. Hence we get:

$$\begin{aligned}
&\int_{\boldsymbol{x}} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}) \cdot \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}) \right| \mathrm{d}\boldsymbol{x} \\
&\quad= \int_{\boldsymbol{x}_1} \int_{\boldsymbol{x}_2} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1) \cdot \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}_2) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}_2) \right| \mathrm{d}\boldsymbol{x_2} \,\mathrm{d}\boldsymbol{x}_1 \\
&\quad= \int_{\boldsymbol{x}_1} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1) \int_{\boldsymbol{x}_2} \cdot \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}_2) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}_2) \right| \mathrm{d}\boldsymbol{x_2} \,\mathrm{d}\boldsymbol{x}_1 \,.
\end{aligned}$$

Recall that $\mathbf{\Pi}\boldsymbol{\mu}_i^* = \boldsymbol{\mu}_i^\dagger$ and $\mathbf{\Pi}\mathbf{\Pi}^T = \boldsymbol{I}_k$, which gives

$$\begin{aligned}
&\int_{\boldsymbol{x}_1} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1) \cdot \int_{\boldsymbol{x}_2} \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\mathbf{\Pi}\boldsymbol{\mu}_i^*}(\boldsymbol{x}_2) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\mathbf{\Pi},\mathbf{\Pi}^T\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}_2) \right| \mathrm{d}\boldsymbol{x}_2 \,\mathrm{d}\boldsymbol{x}_1 \\
&\quad= \int_{\boldsymbol{x}_1} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(x_1) \int_{\boldsymbol{x}_2} \left| \sum_{i=1}^k w_i^* \cdot \mathcal{N}_{\boldsymbol{\mu}_i^\dagger}(\boldsymbol{x}_2) - \sum_{i=1}^k \widehat{w}_i \cdot \mathcal{N}_{\widehat{\boldsymbol{\mu}}_i}(\boldsymbol{x}_2) \right| \mathrm{d}\boldsymbol{x}_2 \,\mathrm{d}\boldsymbol{x}_1 \\
&\quad= \int_{\boldsymbol{x}_1} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1) \int_{\boldsymbol{x}_2} \left| \mathcal{M}_{\theta^\dagger}(\boldsymbol{x}_2) - \mathcal{M}_{\widehat{\theta}(\boldsymbol{x}_2)} \right| \mathrm{d}\boldsymbol{x}_2 \,\mathrm{d}\boldsymbol{x}_1 \\
&\quad= \int_{\boldsymbol{x}_1} \mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1) \cdot O(\epsilon) \,\mathrm{d}\boldsymbol{x}_1 \,,
\end{aligned}$$

where the last line used that $\widehat{\theta}$ is a solution of our system of polynomial inequalities (see Lemma 62). Since $\mathcal{N}_{\overline{\mathbf{\Pi}},0}(\boldsymbol{x}_1)$ is a pdf, it integrates to 1. This shows that the entire integral evaluates to $O(\epsilon)$,

which completes the proof. ■

Finally, we analyze the running time of FITPOLYPROGRAMMULTIVARIATE.

Theorem 64 FITPOLYPROGRAMMULTIVARIATE *runs in time*

$$\left(|V| \cdot k \cdot \log \frac{1}{\zeta}\right)^{O(|V|^2 k^5)} + \widetilde{O}\left(|V| \cdot \frac{k}{\zeta^2}\right).$$

Proof As in the univariate case, we solve our system of polynomial inequalities with Renegar's algorithm SOLVE-POLY-SYSTEM. So in order to obtain a bound on the time complexity, we consider the following quantities:

- As in the univariate case, all polynomials in the predicates of $S_{k\text{-dim},V}$ have degree $O(\log \frac{1}{\zeta})$.
- For each direction $\boldsymbol{v} \in V$, we require $k^{O(k)}$ constraints (see Section 3). Hence the total number of constraints is $|V| \cdot k^{O(k)}$.
- We have $O(k^2)$ free variables (the k vectors of means in $\mathbb{R}^k$ and the k component weights).
- Our system of polynomial inequalities has two levels of quantification. In each level, we have $|V| \cdot k$ bound variables for encoding the $\mathcal{A}_K$-constraints and the order of the breakpoints of the piecewise polynomials.

Substituting these quantities into Fact 5 gives the statement of the theorem. ■

9. Proof of a variant of Conjecture 1 for $k = 2$

In this section, we specialize our results to the case of 2-GMMs with components equally weighted (so weights $1/2$). Define

$$\zeta_{1/k}(\epsilon, k, V) = \inf_{\substack{\theta', \theta'' : \|\mathcal{M}_{\theta'} - \mathcal{M}_{\theta''}\|_1 \geq \epsilon\,, \\ \text{all weights are } 1/k}} \max_{\boldsymbol{v} \in V} \|\mathcal{M}_{\boldsymbol{v}\cdot\theta'} - \mathcal{M}_{\boldsymbol{v}\cdot\theta''}\|_1\,, \tag{14}$$

which is the same as ζ except we focus only on 2-GMMs with weight $1/2$.

Recall our notation for univariate projections of GMMs: If $\boldsymbol{v} \in \mathbb{R}^d$ is a unit vector and θ is a set of GMM parameters, we let $\boldsymbol{v} \cdot \theta$ denote the set of parameters of the univariate marginal distribution induced by projecting the distribution $\mathcal{M}_\theta$ along $\boldsymbol{v}$. That is, $\boldsymbol{v} \cdot \theta$ is the set of parameters with the same weights w_j and precision τ_j as the set of parameters θ, but with means $\langle \boldsymbol{v}, \boldsymbol{\mu}_j \rangle$. For a specific GMM $F = \mathcal{M}_\theta$, we write $F_{\boldsymbol{v}}$ for this univariate projection in order to simplify notation.

This section is devoted to the proof of the following theorem:

Theorem 65 *There is a set of directions V in $\mathbb{R}^2$ of constant size that can be constructed in constant time so that for all $\epsilon > 0$, we have $\zeta_{1/k}(\epsilon, 2, V) = \Omega(\epsilon^3)$.*

We prove this theorem in two parts: first, we show a constant size net has a direction which preserves all mean distances up to some multiplicative constant. We then show that this guarantee suffices to preserve the distance between two GMMs.

9.1. Preserving vector differences via nets

We begin by proving that a random direction preserves the distances between means acceptably and then show that a constant-size net is sufficiently correlated with this random direction. Since these bounds hold for more than two dimensions, we state them for general $\mathbb{R}^d$ as opposed to only $\mathbb{R}^2$. We use the following standard definition of a net, but denote the accuracy parameter with ϕ instead of ϵ in order to avoid confusion with the L_1-accuracy parameter ϵ.

Definition 66 *For any $\phi > 0$, a set of directions V in $\mathbb{R}^d$ is a ϕ-net for the unit sphere if for all unit vectors $\boldsymbol{u}$, there is a $\boldsymbol{v} \in V$ such that $\|\boldsymbol{u} - \boldsymbol{v}\|_2 \leq \phi$.*

Lemma 67 *Let $U = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\} \subset \mathbb{R}^d$ be a set of m unit vectors with $d \geq 2$. Then there exists a unit vector $\boldsymbol{g} \in \mathbb{R}^d$ such that for all $\boldsymbol{u} \in U$ we have $|\langle \boldsymbol{u}, \boldsymbol{g} \rangle| \geq \frac{1}{4m\sqrt{d}}$.*

Proof We show that a random Gaussian vector $\boldsymbol{g'} \sim \mathcal{N}_{\boldsymbol{0}, \frac{1}{d}\boldsymbol{I}}$ gives a "good" direction $\boldsymbol{g} = \frac{\boldsymbol{g'}}{\|\boldsymbol{g'}\|_2}$ with non-zero probability.

First, we consider the inner product between the random vector $\boldsymbol{g'}$ and a single $\boldsymbol{u} \in U$. Due to the rotational symmetry of the multivariate standard normal distribution, it suffices to consider $\langle \boldsymbol{g'}, \boldsymbol{e}_1 \rangle$ for $\boldsymbol{e}_1 = (1, 0, \ldots, 0)^T$. Let $y \sim \mathcal{N}_{0,1}$ and note that g'_1 and $\frac{y}{\sqrt{d}}$ have the same distributions. Hence

$$\begin{aligned}
\mathbb{P}\left(|\langle \boldsymbol{g'}, \boldsymbol{u} \rangle| \leq \frac{1}{2m\sqrt{d}}\right) \;=\; \mathbb{P}\left(|\langle \boldsymbol{g'}, \boldsymbol{e}_1 \rangle| \leq \frac{1}{2m\sqrt{d}}\right) \;&=\; \mathbb{P}\left(|g'_1| \leq \frac{1}{2m\sqrt{d}}\right) \\
&=\; \mathbb{P}\left(|y| \leq \frac{1}{2m}\right) \\
&\leq\; \frac{1}{2\pi m}
\end{aligned}$$

where the last line follows from uniformly bounding the standard normal pdf by its maximum at 0.

Applying a union bound now gives

$$\mathbb{P}\left(\forall \boldsymbol{u} \in U \,:\, |\langle \boldsymbol{g'}, \boldsymbol{u} \rangle| \geq \frac{1}{2m\sqrt{d}}\right) \;\geq\; 1 - m \cdot \mathbb{P}\left(|\langle \boldsymbol{g'}, \boldsymbol{u} \rangle| \leq \frac{1}{2m\sqrt{d}}\right) \;=\; 1 - \frac{1}{2\pi} \;\geq\; \frac{3}{4}\,.$$

It remains to show that the norm of the vector $\boldsymbol{g'}$ is not too large so that the normalization $\frac{\boldsymbol{g'}}{\|\boldsymbol{g'}\|_2}$ does not distort the inner products $|\langle \boldsymbol{g'}, \boldsymbol{u}_i \rangle|$ significantly. Standard tail bounds for the χ^2-distribution show that for any $d \geq 2$ we have $\mathbb{P}(\|\boldsymbol{g'}\|_2 \leq 2) \geq \frac{3}{4}$. So with probability at least $\frac{1}{2}$, the random vector $\boldsymbol{g'}$ satisfies both $\|\boldsymbol{g'}\|_2 \leq 2$ and $|\langle \boldsymbol{g'}, \boldsymbol{u} \rangle| \geq \frac{1}{2m\sqrt{d}}$ for all $\boldsymbol{u} \in U$. Conditioning on this event, we get that

$$|\langle \boldsymbol{g}, \boldsymbol{u} \rangle| \;=\; \frac{1}{\|\boldsymbol{g'}\|_2} |\langle \boldsymbol{g'}, \boldsymbol{u} \rangle| \;\geq\; \frac{1}{4m\sqrt{d}}\,.$$

■

Lemma 68 *Let $U = \{\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m\} \subset \mathbb{R}^d$ be a set of m unit vectors with $d \geq 2$. Moreover, let N be a $\frac{1}{8m\sqrt{d}}$-net for the d-dimensional unit sphere. Then there exists a $\boldsymbol{v} \in N$ such that $|\langle \boldsymbol{u}, \boldsymbol{v} \rangle| \geq \frac{1}{8m\sqrt{d}}$ for all $\boldsymbol{u} \in U$.*

Proof Consider the good direction $\boldsymbol{g} \in \mathbb{R}^d$ from Lemma 67. Since N is a $\frac{1}{8m\sqrt{d}}$-net, we can write the vector $\boldsymbol{g}$ as $\boldsymbol{g} = \boldsymbol{v} + \boldsymbol{e}$ where $\boldsymbol{v} \in N$ and $\boldsymbol{e} \in \mathbb{R}^d$ with $\|\boldsymbol{e}\|_2 \leq \frac{1}{8m\sqrt{d}}$. For any $\boldsymbol{u} \in U$ we then have:

$$\begin{aligned}
\langle \boldsymbol{u}, \boldsymbol{v} \rangle \;=\; \langle \boldsymbol{u}, \boldsymbol{g} - \boldsymbol{e} \rangle \;&=\; \langle \boldsymbol{u}, \boldsymbol{g} \rangle - \langle \boldsymbol{u}, \boldsymbol{e} \rangle \\
&\geq \frac{1}{4m\sqrt{d}} - |\langle \boldsymbol{u}, \boldsymbol{e} \rangle| \\
&\geq \frac{1}{4m\sqrt{d}} - \|\boldsymbol{u}\|_2 \|\boldsymbol{e}\|_2 \\
&\geq \frac{1}{4m\sqrt{d}} - \frac{1}{8m\sqrt{d}} \\
&= \frac{1}{8m\sqrt{d}} \,.
\end{aligned}$$

This completes the proof. ■

As a result, we have:

Corollary 69 *Let V be a $\frac{1}{48\sqrt{2}}$-net for the 2-dimensional unit sphere. Then for all vectors $\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\mu}_3, \boldsymbol{\mu}_4$, there is some $\boldsymbol{v} \in V$ so that $|\langle \boldsymbol{\mu}_i, \boldsymbol{v} \rangle - \langle \boldsymbol{\mu}_j, \boldsymbol{v} \rangle| \geq \frac{1}{48\sqrt{2}} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2$ for all i, j.*

Proof For all $i \neq j$, let $\boldsymbol{u}_{ij} = (\boldsymbol{\mu}_i - \boldsymbol{\mu}_j)/\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2$. There are $\binom{4}{2} = 6$ such $\boldsymbol{u}_{ij}$. By Lemma 68, we know that there is a $\boldsymbol{v} \in V$ so that $|\langle \boldsymbol{v}, \boldsymbol{u}_{ij} \rangle| \geq \frac{1}{48\sqrt{2}}$ for all i, j. Thus for this $\boldsymbol{v}$, we have $|\langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle - \langle \boldsymbol{v}, \boldsymbol{\mu}_j \rangle| \geq \frac{1}{48\sqrt{2}} \|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2$, as claimed. ■

9.2. Preserving mean distances preserves TV distance

We now proceed to our main theorem of this section.

Theorem 70 *Let $F = \mathcal{M}_{1/2,1/2,\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,1,1}$ and $G = \mathcal{M}_{1/2,1/2,\boldsymbol{\nu}_1,\boldsymbol{\nu}_2,1,1}$ be two mixtures of Gaussians in $\mathbb{R}^2$. Moreover, let $\boldsymbol{v}$ be a unit vector with the following property: there is a constant $C > 0$ such that for all $\boldsymbol{a}, \boldsymbol{b} \in \{\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \boldsymbol{\nu}_1, \boldsymbol{\nu}_2\}$, we have $\|\boldsymbol{a} - \boldsymbol{b}\|_2 \leq C|\langle \boldsymbol{v}, \boldsymbol{a} \rangle - \langle \boldsymbol{v}, \boldsymbol{b} \rangle|$. Then if $\|F - G\|_1 \geq \Omega(\epsilon)$, we also have $\|F_v - G_v\|_1 \geq \Omega(\epsilon^3)$, with hidden constants depending only on C.*

Proof For all i, let $\boldsymbol{\mu}'_i = \langle \boldsymbol{v}, \boldsymbol{\mu}_i \rangle$ and $\boldsymbol{\nu}'_i = \langle \boldsymbol{v}, \boldsymbol{\nu}_i \rangle$. First, observe that if there is a matching of the components π so that $|\mu'_i - \nu'_{\pi(i)}| \leq O(\epsilon)$ for $i = 1, 2$, then by the guarantee of the net, this implies that $\|\boldsymbol{\mu}_i - \boldsymbol{\nu}_{\pi(i)}\| \leq O(\epsilon)$, which immediately implies that $\|F - G\|_1 \leq O(\epsilon)$. Hence we may assume that there is a component which is at least $\Omega(\epsilon)$ away from the rest. WLOG, we may assume that $|\mu'_1 - \nu'_i| > \Omega(\epsilon)$, for $i = 1, 2$.

We now split into two cases:

Case 1: $\mu'_1 < \nu'_1 \leq \nu'_2 \leq \mu'_2$. In this case, let $F_1(x) = \mathcal{M}_{1/2,1/2,\mu'_1,\nu'_2,1,1}$, that is, we move the second component of F to the second component of G. Then $F_1(x) \geq F(x)$ for all $x \leq \nu'_2$. However,

$$G(x) - F_1(x) = \frac{1}{2}\mathcal{N}_{\nu'_1,1}(x) - \frac{1}{2}\mathcal{N}_{\mu'_1,1}(x) \,,$$

and thus, on an interval I of length $\Omega(\epsilon)$ with right endpoint at ν_1', we have that $G(x) - F_1(x) \geq \Omega(\epsilon^2)$. Hence, $G(x) - F(x) \geq \Omega(\epsilon^2)$, and therefore $\int |F(x) - G(x)|dx \geq \int_I G(x) - F(x)dx \geq \Omega(\epsilon^3)$.

Case 2: $\mu_1' < \nu_1' \leq \mu_2' \leq \nu_2'$.

Let $F_1(x) = \mathcal{M}_{1/2,1/2,\mu_1',\nu_2',1,1}$, that is, we move the second component of the first mixture to have mean ν_2'. We then clearly have that for all $x \leq \mu_1'$, $F_1(x) \leq F_v(x)$. But we also have that

$$F_1(x) - G_v(x) = \frac{1}{2}\mathcal{N}_{\mu_1',1}(x) - \frac{1}{2}\mathcal{N}_{\nu_1',1}(x)$$

for all $x \leq \mu_1'$. Using $|\mu_1' - \nu_1| \geq \Omega(\epsilon)$, a simple calculation now yields

$$\int_{-\infty}^{\mu_1'} \mathcal{N}_{\mu_1',1}(x) - \mathcal{N}_{\nu_1',1}(x)\,\mathrm{d}x \;\geq\; \Omega(\epsilon)\;.$$

Combining the above inequalities with $\mathcal{N}_{\mu_1',1}(x) - \mathcal{N}_{\nu_1',1}(x) \geq 0$ for all $x \leq \mu_1'$ this gives

$$\begin{aligned}
\|F_v - G_v\|_1 \;&\geq\; \int_{-\infty}^{\mu_1'} |F_v(x) - G_v(x)|\,\mathrm{d}x \\
&=\; \int_{-\infty}^{\mu_1'} F_v(x) - G_v(x)\,\mathrm{d}x \\
&\geq\; \int_{-\infty}^{\mu_1'} F_1(x) - G_v(x)\,\mathrm{d}x \\
&\geq\; w_F \int_{-\infty}^{\mu_1'} \mathcal{N}_{\mu_1',1}(x) - \mathcal{N}_{\nu_1',1}(x)\,\mathrm{d}x \\
&\geq\; w_F \cdot \Omega(\epsilon) \\
&\geq\; \Omega(\epsilon^2)\;,
\end{aligned}$$

which is in fact stronger than we need.

This completes the case analysis and hence also the proof of the theorem. ■

9.3. Putting it together

To complete this subsection, we must simply put the above pieces together in conjunction with the following theorem, which says that ϕ-nets can be constructed efficiently:

Fact 71 (Dadush (2013)) *There is a deterministic algorithm that constructs a ϕ-net N for S^{d-1} such that $|N| \leq (\frac{1}{\phi})^{O(d)}$. Moreover, the algorithm runs in time $(\frac{1}{\phi})^{O(d)}$.*

This immediately gives us as a corollary:

Corollary 72 *There is a set of directions V in $\mathbb{R}^2$ of constant size which can be constructed in constant time so that for all $\epsilon > 0$, we have $\zeta(\epsilon, 2, V) \geq \Omega(\epsilon^3)$.*

10. Putting the multivariate algorithm together

We take a second to recap and put all the ingredients together to finally prove Theorem 47.

Proof [Proof of Theorem 47] We first bound the sample complexity. We have that:

1. COARSEESTIMATESIGMA takes $k+1$ samples.
2. By Theorem 54, RECURSIVESPECTRALPROJECTION with our choice $\delta' = \text{poly}(\delta, 1/k, 1/|V|)$ takes
$$N_2 = \widetilde{O}\left(k^4(d+\log k/\delta)\max\left(\frac{1}{\epsilon^4}, \frac{1}{\zeta^2}\right)\right),$$
samples.
3. FINDAPPROXSUBSPACEANDCOVARIANCE takes
$$N_2 = \widetilde{O}\left(\frac{k^9(d+\log(k/\delta)}{\epsilon^4} + k^{7/2}\frac{d+\log(k/\delta)}{\zeta^2}\right)$$
samples, and finally,
4. FITPOLYPROGRAMMULTIVARIATE takes
$$N_3 = \widetilde{O}\left(\frac{k+\log(|V|/\delta)}{\zeta^2}\right)$$
samples.

Thus overall we take

$$\widetilde{O}\left(k^4(d+\log k/\delta)\max\left(\frac{1}{\epsilon^4}, \frac{1}{\zeta^2}\right) + \frac{k^9(d+\log(k/\delta)}{\epsilon^4} + \frac{k^{7/2}(d+\log(k/\delta)) + \log(|V|/\delta)}{\zeta^2}\right)$$
$$= \widetilde{O}_k\left(\frac{d}{\epsilon^4} + \frac{d+\log(|V|/\delta)}{\zeta^2}\right)$$

samples. We now bound our runtime. We have that:

1. COARSEESTIMATESIGMA takes time $O(k^2)$.
2. By Theorem 54, RECURSIVESPECTRALPROJECTION takes time
$$dk^4(d+\log k/\delta)\max\left(\frac{1}{\epsilon^4}, \frac{1}{\zeta^2}\right)$$
per iteration, and we do at most k iterations.
3. The runtime of a single iteration of FINDAPPROXSUBSPACEANDCOVARIANCE is bounded by the time it takes to run APPROXPCA$(A, k+1, \min(\epsilon^2/k^2, \zeta/k))$ on a matrix A of size $d \times N_2$. We need to do this k times, so the runtime of this is
$$\widetilde{O}\left(\left(\frac{k^{11}(d+\log(k/\delta)}{\epsilon^4} + k^{11/2}\frac{d+\log(k/\delta)}{\zeta^2}\right)\min\left(\frac{k}{\epsilon}, \sqrt{\frac{k}{\zeta}}\right)\cdot d\right).$$

4. Finally, the runtime of the system of polynomial inequalities step is

$$\left(|V| \cdot k \cdot \log \frac{1}{\zeta}\right)^{O(|V|^2 k^5)} + \widetilde{O}\left(|V| \cdot \frac{k}{\zeta^2}\right).$$

Thus the overall runtime is

$$\widetilde{O}_k\left(\frac{|V|}{\zeta^2} + d^2\left(\frac{1}{\epsilon^4} + \frac{1}{\zeta^2}\right) \cdot \min\left(\frac{k}{\epsilon}, \sqrt{\frac{k}{\zeta}}\right) + \left(|V| \log \frac{1}{\zeta}\right)^{O(|V|^2)}\right),$$

which simplifies to the expression in the theorem. ■

10.1. Generalizing to different spherical covariances

We remark that our algorithm naturally generalizes to learn mixture of Gaussians with different covariances if additional structural results about Gaussian mixtures are available. In particular, we require the following conjecture (in addition to Conjecture 1):

Conjecture 73 *Let θ and θ' be two sets of k-GMM parameters in k dimensions. Let $A \in \mathbb{R}^{d \times k}$ be any matrix with orthonormal columns, i.e., an isometric embedding of $\mathbb{R}^k$ into $\mathbb{R}^d$. If $\|\mathcal{M}_\theta - \mathcal{M}_{\theta'}\|_1 \leq \epsilon$ holds, then $\|\mathcal{M}_{A\theta} - \mathcal{M}_{A\theta'}\|_1 < O(\epsilon)$.*

Roughly speaking, this says that if there is a subspace S of dimension k so that all the means of both mixtures are in S, and when projected on this subspace the two mixtures are close, then the two mixtures were close originally. When all the covariances are equal, it is easy to show that this holds, and we tacitly use this fact in the proof of Theorem 63.

Assuming both of our (purely geometric) conjectures, our algorithm naturally generalizes to provably learn arbitrary mixtures of spherical Gaussians. We believe that this is an interesting direction for future work.

10.2. Making the algorithm agnostic

In recent work, Diakonikolas et al. (2016a) give an efficient algorithm for achieving a similar guarantee as RECURSIVESPECTRALPROJECTION and FINDAPPROXSUBSPACEANDCOVARIANCE that runs in polynomial time even in the presence of malicious noise. Roughly speaking, their model is the following:

Definition 74 *Fix a distribution F, and $\epsilon > 0$. An ϵ-corrupted set of samples from $\mathcal{D}$ is generated via the following process: first, draw $\boldsymbol{x}'_1, \ldots, \boldsymbol{x}'_m$ independently from F. Then, an $O(\epsilon)$ fraction of these samples is adversarially corrupted, and the samples are returned in any order.*

Then their results imply the following:

Fact 75 (Diakonikolas et al. (2016a)) *Let $\mathcal{M}_\theta$ be an unknown k-GMM with all covariances equal to I. Fix $\epsilon, \delta > 0$, and let $\boldsymbol{x}_1, \ldots \boldsymbol{x}_n$ be an ϵ-corrupted set of samples from $\mathcal{M}_\theta$, where*

$$n \geq \widetilde{\Omega}_k\left(\frac{d + \log 1/\delta}{\epsilon^2}\right).$$

There is an algorithm AGNOSTICLEARNSUBSPACE$(\epsilon, \tau, \boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$ *that runs in time* $\mathrm{poly}(k, d, 1/\epsilon, \log 1/\delta)$ *and outputs with probability* $1-\delta$ *a valid clustering tree* $\mathcal{T}$ *for* $\mathcal{M}_\theta$ *with error probability* $1 - O(\epsilon)$. *Moreover, for each leaf* ℓ *of* $\mathcal{T}$, *the algorithm outputs a* k*-dimensional subspace* U^ℓ *with projector* $\mathbf{\Pi}^\ell$ *so that if* C^ℓ *is the set of components associated with* ℓ, *then*

$$\sum_\ell \sum_{j \in C^\ell} w_j \left\| \mathcal{N}_{\boldsymbol{\mu}_j, 1} - \mathcal{N}_{\mathbf{\Pi}^\ell \boldsymbol{\mu}_j, 1} \right\|_1 \leq \widetilde{O}_k(\sqrt{\epsilon}) \, .$$

At this point, the algorithm in Diakonikolas et al. (2016a) performs an exhaustive search over the identified subspace U^ℓ and then a tournament to ascertain the component means. This leads to a running time of $(1/\epsilon)^{O(k)}$, i.e., a running time exponential in k. Here, we show at a high level that our techniques for properly learning a GMM can be combined with the approach of Diakonikolas et al. (2016a) and lead to a faster algorithm under Conjecture 1.

Suppose for simplicity that AGNOSTICLEARNSUBSPACE returns a single cluster. Let U be the subspace associated to that cluster, and let its projector be $\mathbf{\Pi}$. Furthermore, let $\mathbf{\Pi}\theta$ denote the set of parameters which are identical to θ except the mean of component j is now at $\mathbf{\Pi}\boldsymbol{\mu}_j$. Our algorithm at this point is in fact unchanged: let V be set of directions within U, and simply run FITPOLYPROGRAMMULTIVARIATE using V and a fresh set of ϵ-corrupted samples[12] from $\mathcal{M}_\theta$ of size at least

$$n \geq \widetilde{\Omega}\left(\frac{k + \log(|V|/\delta)}{\epsilon^2} \right) \, .$$

We then postprocess the output as before.

We sketch the correctness of this algorithm below. First, the correctness of the subroutine FITPOLYPROGRAMMULTIVARIATE requires that the density estimates along each line are still close to the truth. This guarantee is not proved for ESTIMATE-DENSITY in this strong error model. However, for correctness, ESTIMATE-DENSITY requires only that the VC-Inequality holds for the $\mathcal{A}_K$-norm Acharya et al. (2017), and it can be easily verified that the VC-Inequality still holds under this strong adversary up to an additive ϵ error.

Thus, if we let $\widehat{\theta}$ be the set of parameters output by our algorithm, by the same arguments as before, we know that with probability $1-\delta$, for every direction $v \in V$, it holds that $\|\mathcal{M}_{v \cdot \theta} - \mathcal{M}_{v \cdot \widehat{\theta}}\|_1 \leq O(\epsilon)$. By the same arguments as above, if we let ζ^{-1} be so that $\zeta(\zeta^{-1}, k, V) = O(\epsilon)$, we have that

$$\|\mathcal{M}_{\mathbf{\Pi}\theta} - \mathcal{M}_{\widehat{\theta}}\|_1 \leq \zeta^{-1} \, ,$$

and so overall, we have

$$\|\mathcal{M}_\theta - \mathcal{M}_{\widehat{\theta}}\|_1 \leq \widetilde{O}_k(\zeta^{-1} + \sqrt{\epsilon}) \, ,$$

and the runtime of the entire algorithm is

$$\mathrm{poly}(k, d, 1/\epsilon, \log 1/\delta) + \left(|V| \cdot k \cdot \log \frac{1}{\zeta} \right)^{O(|V|^2 k^5)} + \widetilde{O}\left(|V| \cdot \frac{k}{\zeta^2} \right) \, .$$

This has two consequences:

12. Technically as the problem is stated one cannot simply draw an additional set of ϵ-corrupted samples, however one can simply draw a larger set of samples initially and partition them into two groups randomly. It is not hard to show that this suffices, and so we omit the details for simplicity of exposition.

1. By Theorem 65, we know that for mixtures of two Gaussians with weights $1/2$ and equal covariance, we have $\zeta_{1/k}(\epsilon, 2, V) = \Omega(\epsilon^3)$, so that $\zeta^{-1} = \Omega(\epsilon^{1/3})$, and $|V| = O(1)$. These arguments imply that there is an algorithm that runs in time $\mathrm{poly}(k, d, 1/\epsilon, \log 1/\delta) + \widetilde{O}(1)$ for agnostically learning such a mixture up to error $O(\epsilon^{1/3})$.

2. Under Conjecture 1, so that $\zeta = O_k(\epsilon)$ and so $\zeta^{-1} = O_k(\epsilon)$, this algorithm achieves error $\widetilde{O}_k(\sqrt{\epsilon})$ while avoiding the $(1/\epsilon)^{O(k)}$ running time of the algorithm in Diakonikolas et al. (2016a).

11. Numerical experiments for Conjecture 1

In order to test our Conjecture 1, we conduct several numerical experiments in three to five dimensions, i.e., for the cases $k \in \{3, 4, 5\}$ (since Theorem 70 already proves a slightly weaker version of Conjecture 1 for $k = 2$, so we do not explore this case in further detail). In each experiment, we investigate how the k-dimensional L_1-difference between two k-GMMs compares to the L_1-difference on a random one-dimensional projection. More formally, we fix a family of two k-GMMs $\mathcal{M}^{1,\kappa}$ and $\mathcal{M}^{2,\kappa}$ with a "scale parameter" κ and vary κ in order to change the L_1-difference $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$. For each setting of κ, we then estimate the following two quantities via a Monte Carlo simulation:

- The maximum L_1-difference of all directions:

$$\Delta_{\kappa,\max} = \max_{u \in S^{k-1}} \|\mathcal{M}^{1,\kappa}_u - \mathcal{M}^{2,\kappa}_u\|_1 .$$

- The median L_1-difference of all directions:

$$\Delta_{\kappa,\mathrm{median}} = \max\left\{ \delta \;\middle|\; \mathbb{P}_{u \in S^{k-1}}\left[\|\mathcal{M}^{1,\kappa}_u - \mathcal{M}^{2,\kappa}_u\|_1 \geq \delta\right] \geq \frac{1}{2} \right\} .$$

In particular, we are interested in the ratios $\rho_{\kappa,\max} = \Delta_{\kappa,\max}/\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$ and $\rho_{\kappa,\mathrm{median}} = \Delta_{\kappa,\mathrm{median}}/\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$. The quantity $\rho_{\kappa,\max}$ is relevant for Conjecture 1 because a large value of $\rho_{\kappa,\max}$ indicates that there is at least one "good" direction u that achieves an L_1-difference comparable to $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$ on its projection. If such a direction exists, a fine enough net $N \subset S^{k-1}$ should always have a member $u' \in N$ such that u' also achieves a large L_1-difference on its projection. Similarly, a large value of $\rho_{\kappa,\mathrm{median}}$ indicates that there is not only a single good direction but that at least half of the directions capture a non-negligible amount of the L_1-difference between $\mathcal{M}^{1,\kappa}$ and $\mathcal{M}^{2,\kappa}$. Note that the Data Processing Inequality (Corollary 38) gives an upper bound of 1 for both $\rho_{\kappa,\max}$ and $\rho_{\kappa,\mathrm{median}}$.

For each dimension (or equivalently, number of mixture components) $k \in \{3, 4, 5\}$, we conduct experiments with two families of k-GMMs. To limit the number of parameters, we choose I_k as common spherical covariance matrix in all test cases.

- A random instance in which each component mean is chosen as a random unit vector. The component weights are drawn from a Dirichlet distribution with parameters $\alpha_1 = \ldots = \alpha_k = 1$, i.e., the weight vectors are from a uniform distribution over the probability simplex.

- A structured instance in which we plant the following two-dimensional "cross pattern" in orthogonal pairs of dimensions.

  Let $\mu_1^1 = (0,0)$ and $\mu_2^1 = (1,1)$ for the first mixture $\mathcal{M}^1$. Let $\mu_1^2 = (1,0)$ and $\mu_2^2 = (0,1)$ for the second mixture $\mathcal{M}^2$.

  All components have the same weight. This arrangement is interesting because projecting along the standard normal basis vectors $e_1 = (1,0)$ and $e_2 = (0,1)$ collapses the two mixtures so that the projections along e_1 and e_2 always have L_1-difference zero although $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$ can be large. If the number of dimensions / mixture components is odd, we set the weight of the last component to zero.

For two k-GMMs $\mathcal{M}^1$ and $\mathcal{M}^2$ as defined above, we derive a family of k-GMMs $\mathcal{M}^{1,\kappa}$ and $\mathcal{M}^{2,\kappa}$ by multiplying each component mean with κ.

Since there are no closed-form expressions for the L_1-differences between mixtures of Gaussian, we resort to numerical integration in order to approximate the quantities $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$, $\rho_{\kappa,\max}$, and $\rho_{\kappa,\text{median}}$. We implemented our experiments in the Julia programming language (version 0.4) and used the Cubature.jl[13] package for computing approximations of the L_1-differences in one and multiple dimensions. The numerical integration routines in Cubature.jl are based on standard quadrature / cubature algorithms Gentleman (1972); Genz and Malik (1980); Berntsen et al. (1991).

In our experiments, we vary κ from 10^{-1} to 10^{-6}, which results in k-dimensional L_1-differences $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$ that span several orders of magnitude. In all experiments, we set the numerical integration parameters so that the relative precision of our integration approximations is at least 10^{-3}. We estimate the quantities $\Delta_{\kappa,\max}$ and $\Delta_{\kappa,\text{median}}$ by drawing $100,000$ uniformly random directions from the k-dimensional unit sphere and projecting $\mathcal{M}^{1,\kappa}$ and $\mathcal{M}^{2,\kappa}$ onto each direction. Figure 2 shows plots of our results.

For all GMM families and dimensions we considered, the ratios $\rho_{\kappa,\max}$ and $\rho_{\kappa,\text{median}}$ are essentially *constant* over the entire range of the scale parameter τ tested. This indicates that for small learning error tolerances ϵ, we do not require an increasingly larger number of univariate projections in order to "find" the k-dimensional L_1-difference $\epsilon = \|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$. So at least for the cases tested in our experiments, Conjecture 1 holds.

12. An algorithm for agnostically learning GMMs using the L_2-norm

Our algorithms from the previous sections offer strong theoretical guarantees for learning under the L_1-norm, but they rely heavily on subroutines for solving system of polynomial inequalities. While these systems have a small size depending only on k and $\log 1/\epsilon$, they also have a complicated structure involving two levels of quantification. Asymptotically (for large n, or equivalently, small ϵ), the size of the initial density estimation phase dominates. But for practical samples sizes of say $n \leq 10^6$, solving the system of polynomial inequalities is signifiantly more expensive. In order to overcome this barrier, we now present a variant of our univariate algorithm that relies only on *unquantified* systems of polynomial inequalities. While the modified algorithm achieves only a weaker learning guarantee, it also offers a significantly better time complexity: the running time of the GMM fitting phase improves from $(k \log 1/\epsilon)^{O(k^4)}$ to $(k \log 1/\epsilon)^{O(k)}$, which is a crucial step towards making the algorithm practical. Moreover, our experiments demonstrate that our modified algorithm still achieves a good empirical sample complexity, even for the L_1-norm.

13. https://github.com/stevengj/Cubature.jl

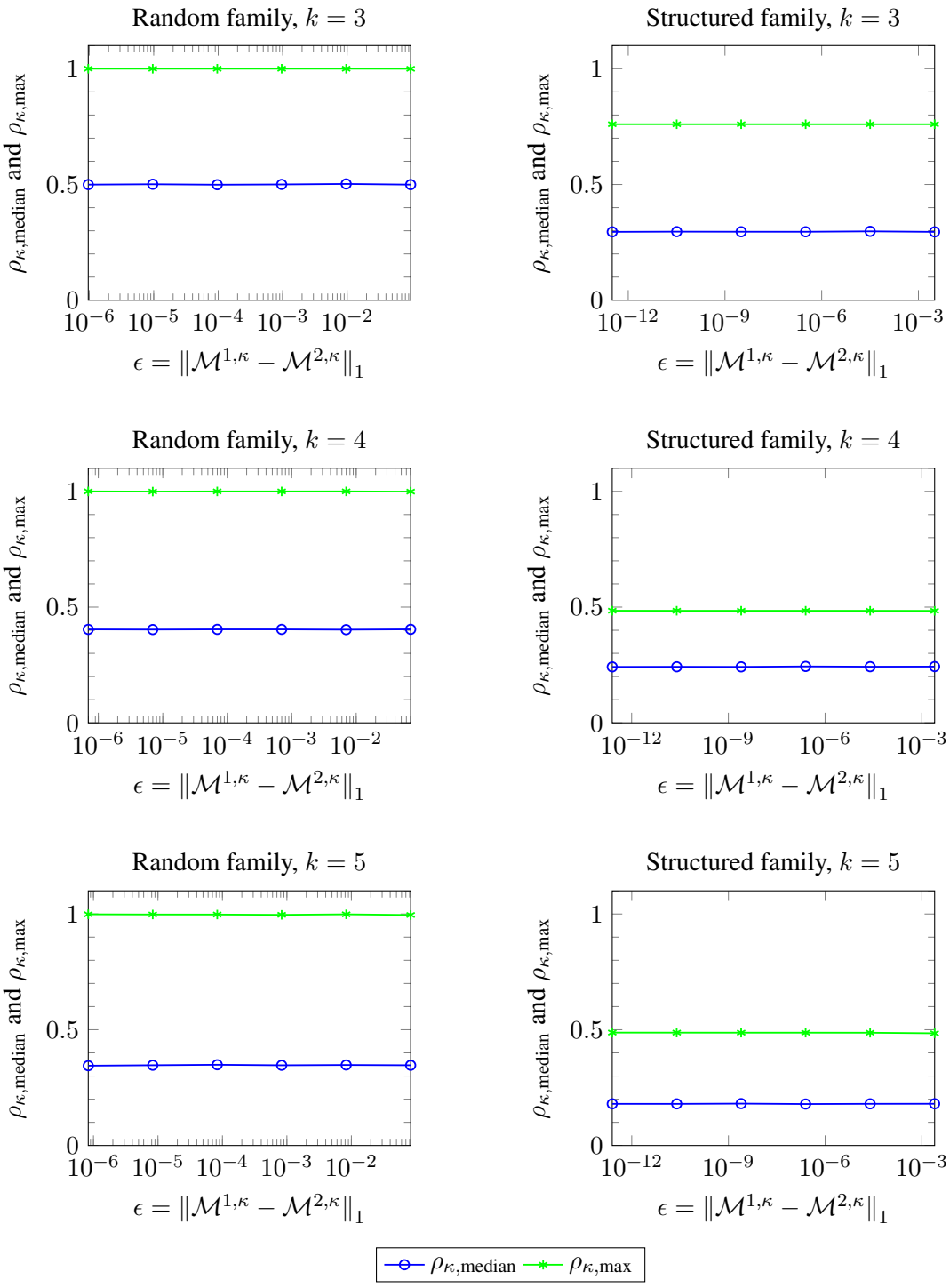

Figure 2: Results of our numerical experiments. All x-axes are for the full k-dimensional L_1-differences $\epsilon = \|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$. The y-axes contain the projected 1-dimensional L_1-difference ratios $\rho_{\kappa,\max}$ and $\rho_{\kappa,\text{median}}$. The data agrees with our Conjecture 1. In particular, the ratios $\rho_{\kappa,\max}$ and $\rho_{\kappa,\text{median}}$ are essentially constant over a wide range of $\|\mathcal{M}^{1,\kappa} - \mathcal{M}^{2,\kappa}\|_1$. This indicates that a net with size independent of $1/\epsilon$ satisfies the conditions of our conjecture.

12.1. Description of the algorithm

We now describe our modified algorithm and give theoretical guarantees. The focus of this section is not to give the best possible guarantees for the problem of properly learning a GMM since we have addressed this question in the previous sections. Instead, our goal is to give meaningful guarantees for an algorithm that we can also study experimentally. In the next section, we then conduct experiments with our algorithm for learning a 2-GMM, both in the agnostic and non-agnostic setting.

As mentioned above, the major obstacle for applying our univariate algorithm on real data is the running time for solving the system of polynomial inequalities. While our system with two levels of quantification is impractical, small *unquantified* systems of polynomial inequalities are within the reach of modern software. In particular, we consider systems of the following form:

$$\begin{aligned} \min \quad & p(\boldsymbol{x}) \text{ s.t.} \\ & q_i(\boldsymbol{x}) \ \Delta_i \ 0, \forall i = 1, \ldots, m \ , \end{aligned}$$

where $\Delta_i \in \{<, \geq, =, \neq, \leq, <\}$ for each $i = 1, \ldots, m$.

To the best of our knowledge, the best theoretical running time for approximately solving such systems is still given by Renegar's algorithm. But as we will see in the next section, our algorithm based on the Mathematica computer algebra system can numerically solve our systems of polynomial inequalities in acceptable time.

The system of polynomial inequalities presented in Section 3 uses quantification to encode the validity of the constraints and to encode the (unknown) breakpoints that achieve the $\mathcal{A}_K$-distance between the density estimate and our shape-restricted polynomial. To massage our system of polynomial inequalities into one without quantifiers, we make the following to changes.

- Instead of enforcing validity of the parameter ordering, we enumerate over all $\exp(O(k))$ possible arrangements for the break points of the optimal shape constrained piecewise polynomial fit.
- To remove the quantifiers used for encoding the $\mathcal{A}_K$-distance, we instead use the L_2-distance. Once the ordering of the intervals is fixed, optimizing the L_2-distance can be written as optimizing an unquantified system of polynomial inequalities.

Thus, for each such possible arrangement, we get a candidate hypothesis. After solving the unquantified system of polynomial inequalities for each arrangement, we can perform a tournament amongst these candidates to choose the best fit, or simply choose the candidate with the smallest error to the density estimate.

12.2. Analysis

Our analysis here requires a somewhat stronger guarantee for the density estimation stage. In particular, we need that the density estimate is not only close in L_1-distance, but also L_∞-distance. While the density estimation procedure of Acharya et al. (2017) does not formally give this guarantee, we find that it usually holds in practice. Moreover, we require a bound on the maximum precision (smallest variance) of any single component, similar to Section 3. However, this is not a severe restriction: if the maximum precision of any single component is too large compared to the

rest, then we can cluster the samples belonging to this component and focus on the remainder. For simplicity of exposition, we do not consider this case.

We follow the notation of Section 3. Let p_{dens} be a given, fixed piecewise polynomial supported on $[-1,1]$ with breakpoints $c_1,\ldots,c_r$. For any $\theta \in \Theta_k$, let the breakpoints of the shape constrained piecewise polynomial associated to θ be $d_1(\theta),\ldots,d_s(\theta)$. Let Φ' be the set of permutations of the variables

$$\{c_1,\ldots,c_r,d_1(\theta),d_s(\theta),-1,1\}\ ,$$

so that the c_i appear in order. For any $\phi=(\phi_1,\ldots,\phi_t)\in\Phi'$, as before, let

$$\mathrm{ordered}'_{p_{\mathrm{dens}},\phi}(\theta) \stackrel{\mathrm{def}}{=} \bigwedge(\phi_i \le \phi_{i+1})\ .$$

Observe that each predicate here is at most a degree four polynomial in the parameters. Moreover, let

$$Q_{p_{\mathrm{dens}},\phi}(\theta)=\sum_{i=1}^{t-1}\int_{\phi_i}^{\phi_{i+1}}\left(p_{\mathrm{dens}}(x)-P_{\epsilon,\theta}(x)\right)^2dx\ .$$

Observe that if θ satisfies $\mathrm{ordered}'_{p_{\mathrm{dens}},\phi}(\theta)$, then $Q_{p_{\mathrm{dens}},\phi}(\theta)$ is just a polynomial in θ. In particular, the problem

$$S(\theta)=\min\ \ Q_{p_{\mathrm{dens}},\phi}(\theta)\quad \text{s.t. } \mathrm{ordered}'_{p_{\mathrm{dens}},\phi}(\theta)$$

is an unquantified system of polynomial inequalities whose solution is the set of parameters θ satisfying the ordering constraints on θ imposed on ϕ with smallest L_2-error to the density estimate. Our full algorithm simply enumerates over all $\phi\in\Phi'$ and for each one finds a θ^*_ϕ by optimizing this problem. At the end, the algorithm returns the θ^*_ϕ with smallest L_1-error to p_{dens} over all arrangements $\phi\in\Phi'$.

The formal pseudocode is given in Algorithm 5.

A full analysis of the algorithm must deal with the fact that we cannot solve the program exactly. Moreover, we must prove that correcting the weights and scaling does not substantially change the output of our algorithm. Since these steps are straightforward, we ignore them for simplicity of exposition.

First, let us bound the runtime of the algorithm:

Theorem 76 LEARN-L2-GMM(k,ϵ,δ) *runs in time*

$$\widetilde{O}\left(\frac{k+\log 1/\delta}{\epsilon^2}\right)+(k\log 1/\epsilon)^{O(k)}\ .$$

Proof As before, the running time of density estimation is

$$\widetilde{O}\left(\frac{k+\log 1/\delta}{\epsilon^2}\right)\ .$$

Hence, it suffices to bound the time the algorithm spends in the GMM fitting part afterwards.

Observe that the running time of our postprocessing is dominated by the time it takes to solve the $\exp(O(k))$ systems of polynomial inequalities. Each system of polynomial inequalities has $O(k)$ unknown variables, $O(k)$ constraints, and polynomials of degree $O(\log 1/\epsilon)$. Hence by Renegar's

Algorithm 5 Algorithm for learning a mixture of Gaussians using L_2-guarantees.

function LEARN-L2-GMM($k, \epsilon, \delta, \gamma$)
  ▷ Density estimation. Only this step draws samples.
  $p'_{\text{dens}} \leftarrow$ ESTIMATE-DENSITY(k, ϵ, δ)
  ▷ Rescaling
  Let p_{dens} be a rescaled and shifted version of p'_{dens} such that the support of p_{dens} is $[-1, 1]$.
  Let α and β be such that $p_{\text{dens}}(x) = p'_{\text{dens}}\left(\frac{2(x-\alpha)}{\beta-\alpha} - 1\right)$
  ▷ Fitting shape-restricted polynomials
  $K \leftarrow 4k$
  for $\phi \in \Phi'$ **do**
    Let $\theta^*_\phi =$ SOLVE-POLY-PROGRAM$(S_\phi, \text{poly}(\epsilon, 1/K), \text{poly}(1/\epsilon, K))$.
    Let $\text{err}_\phi = \int |P_{\epsilon,\theta^*_\phi}(x) - p_{\text{dens}}(x)|dx$.
  Let $\theta^* = \arg\min_{\phi\in\Phi'} \text{err}_\phi$, and let $w_1, \ldots, w_k$ be its weights.
  ▷ Fix the parameters
  for $i = 1, \ldots, k$ **do**
    if $\tau_i \le 0$, set $w_i \leftarrow 0$ and set τ_i to be arbitrary but positive.
  Let $W = \sum_{i=1}^k w_i$
  for $i = 1, \ldots, k$ **do**
    $w_i \leftarrow w_i/W$
  ▷ Undo the scaling
  $w'_i \leftarrow w_i$
  $\mu'_i \leftarrow \frac{(\mu_i+1)(\beta-\alpha)}{2} + \alpha$
  $\tau'_i \leftarrow \frac{\tau_i}{\beta-\alpha}$
  return θ'

algorithm, each individual system can be solved in time $(k \log 1/\epsilon)^{O(k)}$. So the overall running time for the GMM fitting part is

$$\exp(O(k)) \cdot (k \log 1/\epsilon)^{O(k)} = (k \log 1/\epsilon)^{O(k)} .$$

■

Let us now prove correctness of our algorithm.

Theorem 77 *Let f be the underlying distribution, and let θ be so that $\int |f(x) - \mathcal{M}_\theta(x)| \, \mathrm{d}x = \mathrm{OPT}_k$. Let p_{dens} be supported on $[-1, 1]$ so that $\int |p_{\text{dens}}(x) - \mathcal{M}_\theta(x)|d(x) < O(\mathrm{OPT}_k + \epsilon)$ and $\sup_{x\in\mathbb{R}} |p_{\text{dens}}(x) - \mathcal{M}_\theta(x)| \le O(\mathrm{OPT}_k + \xi)$. Then* LEARN-L2-GMM$(k, , \delta)$ *outputs a k-GMM $\mathcal{M}_{\widehat{\theta}}$ so that with probability $1 - \delta$, we have that*

$$\int |f(x) - \mathcal{M}_{\widehat{\theta}}(x)| \, \mathrm{d}x \;\le\; O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \tau_{\max}\epsilon + \xi)} + \mathrm{OPT}_k + \epsilon\right) .$$

Here, $\tau^2_{\max} = \max_{i=1}^k \tau_i^2$, where $\tau_1, \ldots, \tau_k$ are the precisions for the components of $\mathcal{M}_\theta$.

Let us briefly pause to explain this theorem. When $\tau_{\max}$ and ξ are both reasonable (i.e., $\tau_{\max} = O(1)$ and $\xi = O(\epsilon)$), this expression simplifies to $\int |f - \mathcal{M}_\theta| \, \mathrm{d}x \leq O(\mathrm{OPT}_k + \epsilon)$, which is our typical guarantee.

Proof We proceed in two steps. First, we show that there must be a solution p with small L^2 error. Then, we show that this solution also has small variational distance.

Observe that if $\widetilde{\mathcal{P}}_\epsilon(x)$ is as in Definition 7, from Taylor's theorem we have that $\sup_{x\in\mathbb{R}} |\widetilde{\mathcal{P}}_\epsilon(x) - \mathcal{N}(x)| \leq \epsilon$. In particular, since $P_{ve,\theta}$ is the mixture of the scaled Taylor expansions of each component of degree $O(\log 1/\epsilon)$, this implies that

$$\sup_{x\in\mathbb{R}} |P_{\epsilon,\theta}(x) - \mathcal{M}_\theta(x)| \leq \sum_{i=1}^{k} w_i \tau_i |\widetilde{\mathcal{P}}_\epsilon(\tau_i(x-\mu_i)) - \mathcal{N}(\tau_i(x-\mu_i))| \leq \tau_{\max}\epsilon \; .$$

Hence by assumption, we have $\sup_{x\in\mathbb{R}} |p_{\mathrm{dens}}(x) - P_{\epsilon,\theta}(x)| \leq O(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)$. Thus, by Hölder's inequality we must have that

$$\begin{aligned}\int (p_{\mathrm{dens}}(x) - P_{\epsilon,\theta}(x))^2 dx &\leq \int |p_{\mathrm{dens}}(x) - P_{\epsilon,\theta}(x)| dx \cdot \sup_{x\in\mathbb{R}} |p_{\mathrm{dens}}(x) - P_{\epsilon,\theta}(x)| \\ &\leq O\left((\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)\right) \; .\end{aligned}$$

Let $\phi \in \Phi'$ be the ordering satisfied by the breakpoints of p_{dens} and $P_{\epsilon,\theta}$. Let θ^*_ϕ be the output of Renegar's algorithm for our system of polynomial inequalities this ϕ. By the above, we know that $\int (p_{\mathrm{dens}}(x) - P_{\epsilon,\theta^*_\phi})^2 dx \leq O\left((\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)\right)$.

We now seek to show that this θ^*_ϕ must have small variational distance to p_{dens} and f. Indeed, by Jensen's inequality, we have that

$$\left(\int_{-1}^{1} \left| p_{\mathrm{dens}}(x) - P_{\epsilon,\theta^*_\phi} \right| dx\right)^2 \leq 2(p_{\mathrm{dens}}(x) - P_{\epsilon,\theta^*_\phi})^2 dx \; ,$$

and hence

$$\int_{-1}^{1} \left| p_{\mathrm{dens}}(x) - P_{\epsilon,\theta^*_\phi} \right| dx \leq O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)}\right) \; .$$

Moreover, since $\int_{-1}^{1} p_{\mathrm{dens}}(x) dx = 1$ and $p_{\mathrm{dens}}(x) \geq 0$, this implies that

$$\int_{-1}^{1} |P_{\epsilon,\theta^*_\phi}| dx \geq 1 - O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)}\right) \; .$$

Since

$$\int |P_{\epsilon,\theta^*_\phi}| dx \leq \int |P_{\epsilon,\theta^*_\phi} - \mathcal{M}_{\theta^*_\phi}| dx + \int |\mathcal{M}_{\theta^*_\phi}| dx \leq 1 + \epsilon \; ,$$

this implies that

$$\int_{x\notin[-1,1]} |P_{\epsilon,\theta^*_\phi}| dx \leq O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)}\right) + \epsilon \; .$$

Thus altogether we must have that

$$\int \left| p_{\mathrm{dens}}(x) - P_{\epsilon,\theta^*_\phi} \right| dx \leq O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)}\right) + \epsilon \; ,$$

and hence by the triangle inequality, we have

$$\int \left| f - \mathcal{M}_{\theta^*_\phi} \right| dx \leq O\left(\sqrt{(\mathrm{OPT}_k + \epsilon)(\mathrm{OPT}_k + \xi + \tau_{\max}\epsilon)} + \mathrm{OPT}_k + \epsilon \right) ,$$

as claimed. ■

13. Experiments with our univariate algorithm

We now investigate the emprirical performance of our algorithm proposed in the previous section. We emphasize that the experiments here are only a preliminary evaluation of our algorithm with a focus on empirical sample complexity. We believe that the running time of our algorithm can be improved significantly by going beyond the black-box solver provided in Mathematica.

Before we start with a description of our experiments, we give a brief overview of our implementation. Our algorithm consists of three parts:

1. The piecewise polynomial density estimation algorithm of Acharya et al. (2017). The algorithm is written in a combination of Python and C++.

2. A Python program that produces a set of candidate arrangements for a given density estimate. For each arrangement, the Python program also produces a system of polynomial inequalities in the form of a Mathematica program.

3. A set of automatically generated Mathematica progams (one per arrangement) for finding the GMM parameters. Each program first carries out the symbolic computations for producing the relevant error polynomials and then solves the corresponding system of polynomial inequalities. We use the Mathematica function NMinimize to numerically solve the systems of polynomial inequalities.

For the error polynomials, we use a degree-6 Chebyshev approximation. We found this approximation sufficient to achieve a good learning error. For the improper density estimate in the first stage of our algorithm, we use a piecewise polynomial with 5 pieces and degree 5.

We conduct experiments with our algorithm on the task of properly learning a univariate 2-GMM (see Figure 3). We consider two variants of this task: in the noiseless / non-agnostic version, the samples come from a true 2-GMM. In the noisy / agnostic version, we have perturbed the 2-GMM by making its left tail slightly heavier (the probability mass of the noise is 0.05). The noisy version is significantly more challenging since the learning algorithm has to be robust to the noise in the distribution.

We compare our algorithm to three baselines:

- The **improper learning** algorithm of Acharya et al. (2017). This is an interesting comparision for two reasons: first, the algorithm of Acharya et al. (2017) offers (nearly) optimal ttime and sample complexity for the task of *improperly* learning a 2-GMM. Moreover, the improper algorithm is the first step in our proper learning algorithm. So comparing the performance of the improper algorithm with our proper algorithm allows us to study the impact of our regularization in the form of proper learning.

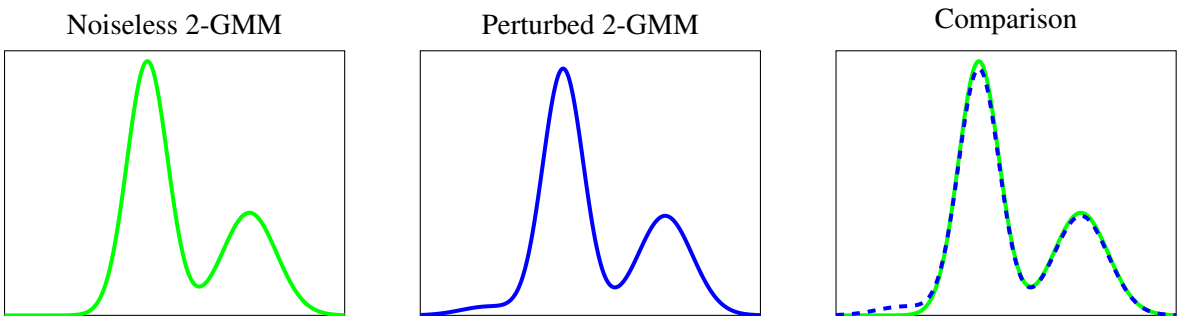

Figure 3: Our test distributions. Left plot: the noiseless 2-GMM. Middle plot: the slightly perturbed 2-GMM. Note that the left tail is slightly heavier. The total probability mass of the noise is 0.05. Right plot: the two densities laid on top of each other.

- **Kernel density estimation**, which is a standard technique for density estimation / improper learning. We use the Epanechnikov kernel because it offers the best guarantees and Silverman's rule for bandwidth selection (we also tried other bandwidths but found Silverman's rule to work best). We use the implementation of kernel density estimation provided by scikit-learn.
- The **Expectation-Maximization** (EM) algorithm, which is a popular algorithm for learning GMMs. The EM algorithm maximizes the likelihood of the data but is not guaranteed to reach a global maximum. Instead, the EM algorithm is typically run with many random initializations to find a good solution. We use 1000 random initializations and verified that more random initializations did not improve the performance of the EM algorithm in our tests significantly. We use the implementation of the EM algorithm provided by scikit-learn.

All experiments were conducted on a laptop computer with a 2.8 GHz Intel Core i7 CPU and 16 GB of RAM. For the Mathematica part of our algorithm, we used Mathematica 10.4. Every data point in the following experiments was averaged over 10 independent trials.

13.1. Noiseless experiments

In this set of experiments, we compare the empirical sample complexity and running time of the four algorithms. We varied the sample size from $n = 10^3$ to $n = 10^5$ and recorded the running times and L_1-difference between the ground truth GMM and the hypothesis produced by the algorithm. See Figure 4 for the corresponding results.

The experiments show that our proper learning algorithm has a better empirical sample complexity than both kernel density estimation and the improper learning algorithm. Moreover, the sample complexity is competitive with the EM algorithm. Interestingly, our postprocessing of the improper density estimate significantly improves the average L_1-error: for $n = 10^5$ samples, our proper learning algorithm has an average L_1-error of 0.006, while the improper algorithm achieves only an average error of 0.015, i.e., our proper GMM fitting improves the learning accuracy by more than a factor of two.

In terms of time complexity, solving the system of polynomial inequalities with a black-box method occurs a significant overhead in terms of running time. For $n = 10^5$ samples, our algorithm takes about 45 minutes to complete. As mentioned above, we believe that is possible to significantly improve this running time, which is an interesting direction for future work. As predicted by our theoretical analysis, the running time of our algorithm is essentially independent of n for the values of n studied here because the time spent on solving the systems of polynomial inequalities dominates the overall running time.

13.2. Experiments with noise

Next, we study the learning accuracy of the four algorithms when the samples come from a slightly perturbed 2-GMM as opposed to a true 2-GMM. As before, we varied the sample size from $n = 10^3$ to $n = 10^5$ and recorded the learning errors (we omit another runnig time plot because the running times are comparable to the noiseless case above). This time, we record two different error quantities: (i) The L_1-distance to the density of the perturbed 2-GMM we draw samples from. (ii) The L_1-distance to the density of the original (unperturbed) 2-GMM. The improper learning algorithms like kernel density estimation and the algorithm of Acharya et al. (2017) aim to minimize the first quantity. While our guarantees in the previous sections are also with respect to the density from which we draw samples, it is also interesting to see whether our algorithm can "de-noise" the distribution and produce an estimate that is close to the original, unperturbed 2-GMM. See Figure 5 for the corresponding results.

The results show several points. While the EM algorithm offered the best learning accuracy in the noiseless case, it fails to provide accurate approximations of the underlying distribution in the noisy case. It is well known that the EM algorithm is not robust to outliers, and our experiments confirm this point. The EM algorithm fails because samples in the tail of a Gaussian have a very small likelihood. As a result, the EM algorithm decides to shift a significant fraction of a GMM component towards the outliers and increases the variance of this component (see Figure 1 in the introduction). This leads to a larger error in L_1-norm, both to the perturbed and unperturbed 2-GMM density. We remark that we ran the EM algorithm with a large number of random initializations and observed no significant difference between $1,000$ and $10,000$ restarts. Hence we belive that the worse performance of the EM algorithm is due to the unsuitable objective function and not its failure to find a good objective value.

For approximating the perturbed 2-GMM density, the improper learning methods (kernel density estimation and the algorithm of Acharya et al. (2017)) offer the best learning accuracy for large n. This is because the perturbed 2-GMM cannot be approximated better than OPT_2 by our proper learning algorithm. It is worth noting that for $n = 10^5$, our algorithm achieves an L_1-error to the perturbed density of about 0.05, i.e., almost exactly the size of the perturbation from the original 2-GMM. For small $n = 10^3$, our algorithm still improves over the improper learning algorithm and is competitive with kernel density estimation.

Finally, we consider the L_1-error to the original, unperturbed 2-GMM. Here, our algorithm offers the best approximation by a significant margin. For $n = 10^5$, our algorithm achieves an average L_1-error of 0.024, while the improper algorithm and kernel density estimation achieve only about 0.056. An approximation of 0.05 is a natural bottleneck for the improper algorithms in this case because the total mass of the perturbation is 0.05. Interestingly, our algorithm is able

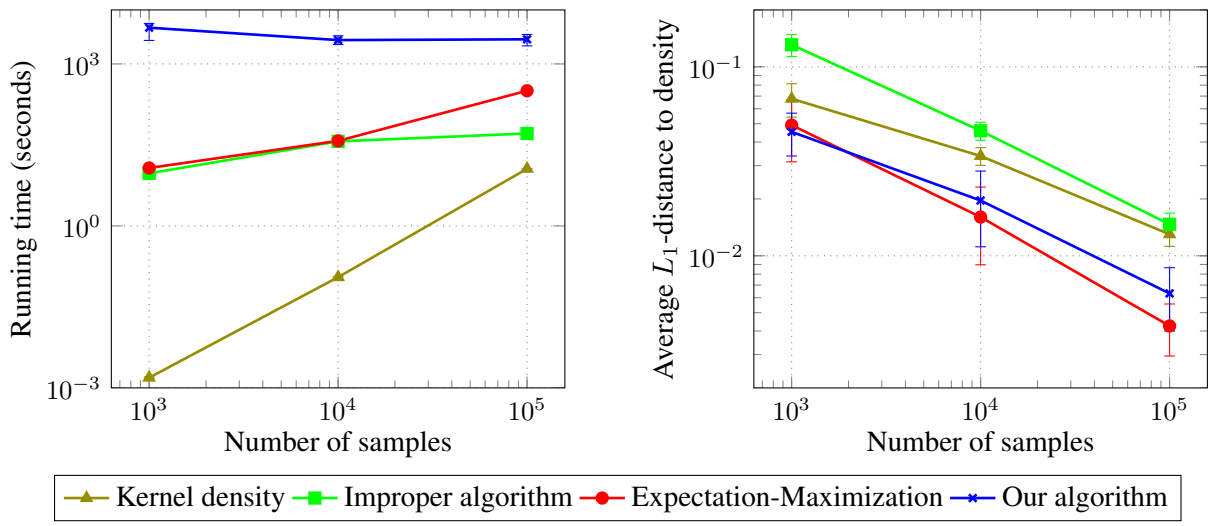

Figure 4: Results for learning a 2-GMM without noise / non-agnostically. The left plot shows the running time of the four algorithms. As predicted by our theory, the running time of our proper learning algorithm is essentially independent of the sample size in this regime because the running time is dominated by the time spent on solving our systems of polynomial inequalities. The right plot shows the learning error. Our algorithm improves over both kernel density estimation and the improper learning baseline. Moreover, our algorithm is competitive with the EM algorithm.

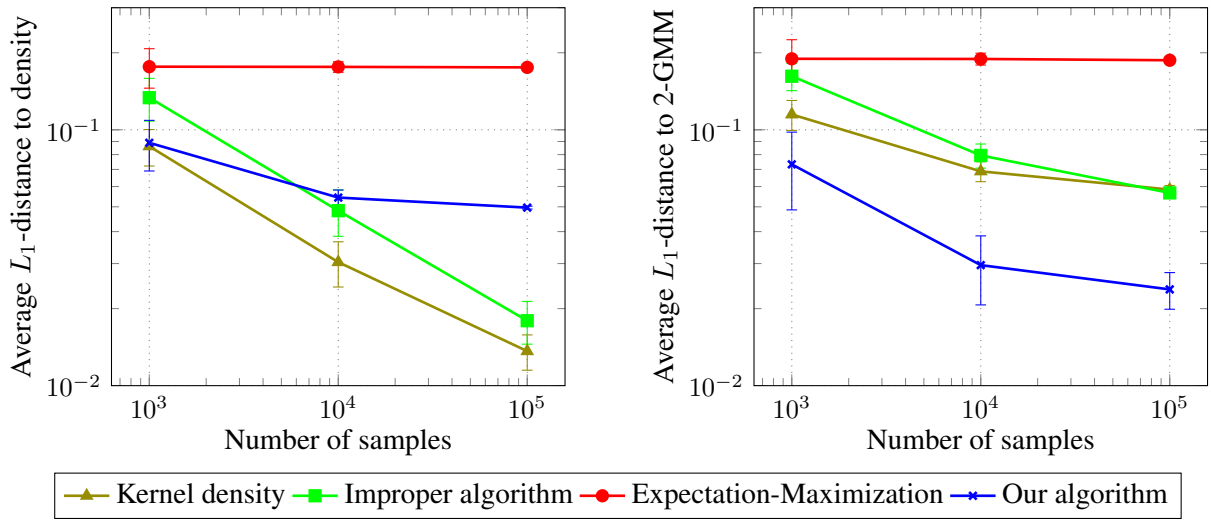

Figure 5: Results for learning a perturbed 2-GMM / agnostic learning (the total probability mass of the perturbation is 0.05, see Figure 3). The left plot shows the L_1-learning error with respect to the perturbed density. The right plot shows the L_1-learning error with respect to the unperturbed 2-GMM. In both cases, the EM algorithm does not produce an accurate hypothesis. In contrast, our algorithm is competitive up to the natural noise floor when the error is measured w.r.t. the perturbed density. When the error is measured w.r.t. the original 2-GMM, our algorithm succeeds to "denoise" the perturbed distribution.

to approximate the original 2-GMM better than this perturbation, i.e., the algorithm succceeds in "denoising" the perturbed distribution.

Acknowledgments

We thank Jayadev Acharya, Ilias Diakonikolas, Piotr Indyk, Gautam Kamath, Ankur Moitra, Cameron Musco, Christopher Musco, Ilya Razenshteyn, Rocco Servedio, Ananda Theertha Suresh, and Santosh Vempala for helpful discussions.

References

Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda Theertha Suresh. Near-optimal-sample estimators for spherical Gaussian mixtures. In *NIPS*, 2014.

Jayadev Acharya, Ilias Diakonikolas, Jerry Li, and Ludwig Schmidt. Sample-optimal density estimation in nearly-linear time. 2017.

Dimitris Achlioptas and Frank McSherry. On spectral learning of mixtures of distributions. In *Learning Theory*, volume 3559 of *Lecture Notes in Computer Science*, pages 458–469. 2005.

Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James R. Voss. The more, the merrier: the blessing of dimensionality for learning large Gaussian mixtures. In *COLT*, 2014.

Sanjeev Arora and Ravi Kannan. Learning mixtures of arbitrary Gaussians. In *STOC*, 2001.

Sivaraman Balakrishnan, Martin J. Wainwright, and Bin Yu. Statistical guarantees for the EM algorithm: From population to sample-based analysis. arXiv:1408.2156, 2014.

Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. In *FOCS*, 2010.

Jarle Berntsen, Terje O. Espelid, and Alan Genz. An adaptive algorithm for the approximate calculation of multiple integrals. *ACM Transactions on Mathematical Software*, 17(4):437–451, 1991.

Aditja Bhaskara, Ananda Theertha Suresh, and Morteza Zadimoghaddam. Sparse solutions to non-negative linear systems and applications. In *AISTATS*, 2015.

Aditya Bhaskara, Moses Charikar, Ankur Moitra, and Aravindan Vijayaraghavan. Smoothed analysis of tensor decompositions. In *STOC*, 2014.

Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., 1995.

S. Charles Brubaker. Robust PCA and clustering in noisy mixtures. In *SODA*, 2009.

S. Charles Brubaker and Santosh Vempala. Isotropic PCA and affine-invariant clustering. In *FOCS*, 2008.

Siu-On Chan, Ilias Diakonikolas, Rocco Servedio, and Xiaorui Sun. Learning mixtures of structured distributions over discrete domains. In *SODA*, 2013.

Siu-On Chan, Ilias Diakonikolas, Rocco Servedio, and Xiaorui Sun. Efficient density estimation via piecewise polynomial approximation. In *STOC*, 2014.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*. Wiley-Interscience, 2006. ISBN 0471241954.

Daniel Dadush. A deterministic polynomial space construction for eps-nets under any norm. *CoRR*, abs/1311.6671, 2013.

Sanjoy Dasgupta. Learning mixtures of Gaussians. In *FOCS*, 1999.

Sanjoy Dasgupta and Leonard J. Schulman. A probabilistic analysis of em for mixtures of separated, spherical gaussians. *Journal of Machine Learning Research*, 8:203–226, 2007.

Constantinos Daskalakis and Gautam Kamath. Faster and sample near-optimal algorithms for proper learning mixtures of gaussians. In *COLT*, 2014.

Constantinos Daskalakis, Christos Tzamos, and Manolis Zampetakis. Ten steps of em suffice for mixtures of two gaussians. In *NIPS 2016 Workshop on Non-Convex Optimization for Machine Learning*, 2016. URL https://arxiv.org/abs/1609.00368.

Luc Devroye and Gabor Lugosi. *Combinatorial methods in density estimation*. Springer Series in Statistics, Springer, 2001.

Ilias Diakonikolas. Learning structured distributions. In Peter Bühlmann, Petros Drineas, Michael J. Kane, and Mark Van Der Laan, editors, *Handbook of Big Data*. Chapman and Hall/CRC, 2016.

Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In *FOCS*, 2016a.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Properly learning poisson binomial distributions in almost polynomial time. In *COLT*, 2016b.

Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. *arXiv preprint arXiv:1611.03473*, 2016c.

Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Learning multivariate log-concave distributions. *CoRR*, abs/1605.08188, 2016d. URL http://arxiv.org/abs/1605.08188.

Jon Feldman, Rocco A. Servedio, and Ryan O'Donnell. PAC learning axis-aligned mixtures of Gaussians with no separation assumption. In *COLT*, 2006.

Jon Feldman, Ryan O'Donnell, and Rocco A. Servedio. Learning mixtures of product distributions over discrete domains. *SIAM Journal on Computing*, 37(5):1536–1564, 2008.

Yoav Freund and Yishay Mansour. Estimating a mixture of two product distributions. In *COLT*, 1999.

Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of gaussians in high dimensions. In *STOC*, 2015.

W. Morven Gentleman. Implementing clenshaw-curtis quadrature, i methodology and experience. *Communications of the ACM*, 15(5):337–342, 1972.

Alan C. Genz and Arham A. Malik. Remarks on algorithm 006: An adaptive algorithm for numerical integration over an n-dimensional rectangular region. *Journal of Computational and Applied Mathematics*, 6(4):295 – 302, 1980.

Moritz Hardt and Eric Price. Tight bounds for learning a mixture of two Gaussians. In *STOC*, 2015.

Daniel Hsu and Sham M. Kakade. Learning mixtures of spherical Gaussians: Moment methods and spectral decompositions. In *ITCS*, 2013.

Arian Maleki Ji Xu, Daniel Hsu. Global analysis of expectation maximization for mixtures of two gaussians. In *NIPS*, 2016.

Chi Jin, Yuchen Zhang, Sivaraman Balakrishnan, Martin J. Wainwright, and Michael I. Jordan. Local maxima in the likelihood of gaussian mixture models: Structural results and algorithmic consequences. In *NIPS*, 2016.

Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two Gaussians. In *STOC*, 2010.

Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. The spectral method for general mixture models. *SIAM Journal on Computing*, 38(3):1141–1156, 2008.

Michael J. Kearns, Robert E. Schapire, and Linda M. Sellie. Toward efficient agnostic learning. *Machine Learning*, 17(2-3):115–141, 1994.

Arlene K. H. Kim and Richard J. Samworth. Global rates of convergence in log-concave density estimation. *Annals of Statistics*, 44(6):2756–2779, 12 2016.

Sergey Levine, Chelsea Finn, Trevor Darrell, and Pieter Abbeel. End-to-end training of deep visuomotor policies. *Journal of Machine Learning Research*, 17(1):1334–1373, 2016.

Ankur Moitra. Algorithmic aspects of machine learning. http://people.csail.mit.edu/moitra/docs/bookex.pdf, 2014.

Ankur Moitra and Gregory Valiant. Settling the polynomial learnability of mixtures of Gaussians. In *FOCS*, 2010.

Cameron Musco and Christopher Musco. Stronger and faster approximate singular value decomposition via the block lanczos method. In *NIPS*, 2015.

Karl Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 185:71–110, 1894.

James Renegar. On the computational complexity and geometry of the first-order theory of the reals. Part i: Introduction. Preliminaries. The geometry of semi-algebraic sets. the decision problem for the existential theory of the reals. *Journal of Symbolic Computation*, 13(3):255 – 299, 1992a.

James Renegar. On the computational complexity of approximating solutions for real algebraic formulae. *SIAM Journal on Computing*, 21(6):1008–1025, 1992b.

Aleksandr F. Timan. *Theory of Approximation of Functions of a Real Variable*. Pergamon, New York, 1963.

Timo Tossavainen. On the zeros of finite sums of exponential functions. *Australian Mathematical Society Gazette*, 33(1):47 – 50, 2006.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 1st edition, 2008. ISBN 0387790519, 9780387790510.

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

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.