Chapter 6

Gaussian Mixture Models

In this chapter we will study *Gaussian mixture models* and clustering. The basic problem is, given random samples from a mixture of k Gaussians, we would like to give an efficient algorithm to learn its parameters using few samples. If these parameters are accurate, we can then cluster the samples and our error will be nearly as accurate as the Bayes optimal classifier.

6.1 History

The problem of learning the parameters of a mixture of Gaussians dates back to the famous statistician Karl Pearson (1894) who was interested in biology and evolution. In fact, there was a particular species of crab called the Naples crab that inhabited the region around him. He took thousands of samples from this population and measured some physical characteristic of each sample. He plotted the frequency of occurrence, but the resulting density function surprised him. He expected that it would be Gaussian, but in fact it was not even symmetric around its maximum value. See Figure 6.1. He hypothesized that maybe the Naples crab was not one species but rather two, and that the density function he observed could be explained as a mixture of Gaussians.

In this remarkable study Pearson introduced the *method of moments*. His basic idea was to compute empirical moments from his samples, and use each of these empirical moments to set up a system of polynomial equations on the parameters of the mixture. He solved this system *by hand*! In fact, we will return to his basic approach later in this unit.

Basics

Here we formally describe the problem of learning mixtures of Gaussians. Recall that for a univariate Gaussian we have that its density function is given by:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left\{\frac{-(x-\mu)^2}{2\sigma^2}\right\}$$

The density of a multidimensional Gaussian in $\mathbb{R}^n$ is given by:

$$\mathcal{N}(\mu, \Sigma) = \frac{1}{(2\pi)^{n/2} det(\Sigma)^{1/2}} exp\left\{\frac{-(x-\mu)^\top \Sigma^{-1}(x-\mu)}{2}\right\}$$

Here Σ is the covariance matrix. If $\Sigma = I_n$ and $\mu = \vec{0}$ then the distribution is just: $\mathcal{N}(0,1) \times \mathcal{N}(0,1) \times ... \times \mathcal{N}(0,1)$.

A mixture of two Gaussians is a distribution whose density function is:

$$F(x) = w_1 F_1(x) + (1 - w_1) F_2(x)$$

where F_1 and F_2 are Gaussians. We can generate a random sample as follows: with probability w_1 we output a random sample from F_1, and otherwise we output a random sample from F_2. Our basic problem is to learn the parameters that describe the mixture given random samples from F. We note that we will measure how good an algorithm is by both its *sample complexity* and its running time.

Method of Moments

Pearson used the *method of moments* to fit a mixture of two Gaussians to his data. The moments of a mixture of Gaussians are themselves a polynomial in the unknown parameters, which we will denote by M_r.

$$\mathop{\mathbb{E}}_{x \leftarrow F_1(x)}[x^r] = M_r(\mu, \sigma^2)$$

Then we can write

$$\mathop{\mathbb{E}}_{x \leftarrow F(x)}[x^r] = w_1 M_r(\mu_1, \sigma_1^2) + (1 - w_1) M_r(\mu_2, \sigma_2^2) = P_r(w_1, \mu_1, \sigma_1, \mu_2^2, \sigma_2^2)$$

And hence the r^{th} raw moment of a mixture of two Gaussians is itself a degree $r+1$ polynomial (P_r) in the unknown parameters.

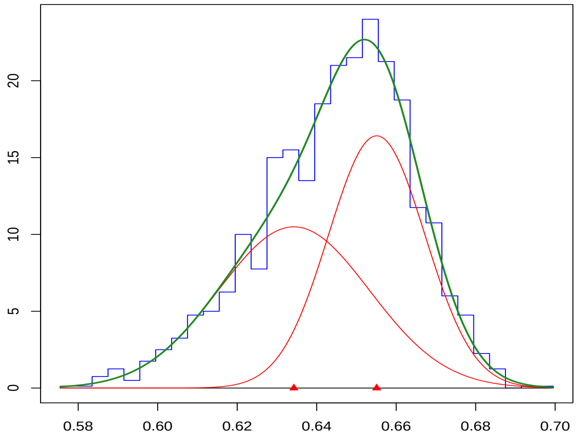

Figure 6.1: A fit of a mixture of two univariate Gaussians to the Pearson's data on Naples crabs, created by Peter Macdonald using R

Pearson's Sixth Moment Test: We can estimate $\mathbb{E}_{x \leftarrow F}[x^r]$ from random samples: Let S be our set of samples. Then we can compute:

$$\widetilde{M}_r = \frac{1}{|S|} \sum_{x \in S} x^r$$

And given a polynomial number of samples (for any $r = O(1)$), $\widetilde{M}_r$ will be additively close to $\mathbb{E}_{x \leftarrow F(x)}[x^r]$. Pearson's approach was:

- Set up a system of polynomial equations

$$\left\{ P_r(w_1, \mu_1, \sigma_1, \mu_2^2, \sigma_2^2) = \widetilde{M}_r \right\}, \; r = 1, 2, ...5$$

- Solve this system. Each solution is a setting of all five parameters that explains the first five empirical moments.

Pearson solved the above system of polynomial equations *by hand*, and he found a number of candidate solutions. Each solution corresponds to a simultaneous setting of the parameters so that the moments of the mixture would match the empirical moments. But how can we choose among these candidate solutions? Some of the solutions were clearly not right; some had negative values for the variance, or a value for the mixing weight not in the range $[0, 1]$. But even after eliminating these solutions, Pearson was still left with more than one candidate. His approach was to choose the root whose prediction is closest to the empirical sixth moment $\widetilde{M}_6$. This is called the *sixth moment test*.

Expectation-Maximization

Much of modern statistics instead focuses on the *maximum likelihood estimator*, which would choose to set the parameters to as to maximize the probability that the mixture would generate the observed samples. Unfortunately, this estimator is NP-hard to compute [18]. The popular alternative is known as *expectation-maximization* and was introduced in a deeply influential paper of Dempster, Laird, Rubin [50]. The basic approach is to repeat the following steps until convergence:

- For each $x \in S$, calculate the posterior probability:

$$w_1(x) = \frac{w_1 F_1(x)}{w_1 F_1(x) + (1 - w_1) F_2(x)}$$

- Update the mixing weights:

$$w_1 \leftarrow \frac{\sum_{x \in S} w_1(x)}{|S|}$$

- Re-estimate the parameters:

$$\mu_i \leftarrow \frac{\sum_{x \in S} w_i(x) x}{\sum_{x \in S} w_i(x)}, \ \Sigma_i \leftarrow \frac{\sum_{x \in S} w_i(x)(x - \mu_i)(x - \mu_i)^\top}{\sum_{x \in S} w_i(x)}$$

This approach gets stuck in local optima, and is in general quite sensitive to how it is initialized (see e.g. [105]).

6.2 Clustering-Based Algorithms

Our basic goal will be to give algorithms that provably compute the true parameters of a mixture of Gaussians, given a polynomial number of random samples. This question was introduced in the seminal paper of Dasgupta [45], and the first generation of algorithms focused on the case where the components of the mixture have essentially no "overlap". The next generation algorithms are based on algebraic ideas, and avoid clustering altogether.

Before we proceed, we will discuss some of the counter-intuitive properties of high-dimensional Gaussians. To simplify the discussion, we will focus on spherical Gaussians $\mathcal{N}(\mu, \sigma^2 I)$ in $\mathbb{R}^n$.

Fact 6.2.1 *The maximum value of the density function is at $x = \mu$.*

Fact 6.2.2 *Almost all of the weight of the density function has* $\|x-\mu\|_2^2 = \sigma^2 n \pm \sigma^2\sqrt{n \log n}$

These facts seem to be inconsistent, but the explanation is that the surface area increases faster as the radius R increases than the value of the density function decreases, until we reach $R^2 \approx \sigma^2 n$. Hence we should think about a high-dimensional spherical Gaussian as being a ball of radius $\sigma\sqrt{n}$ with a thin shell.

Dasgupta [45] – $\widetilde{\Omega}(\sqrt{n})$ Separation

Dasgupta gave the first provable algorithms for learning mixtures of Gaussians, and required that $\|\mu_i - \mu_j\|_2 \geq \widetilde{\Omega}(\sqrt{n}\sigma_{max})$ where σ_{max} is the maximum variance of any Gaussian in any direction (e.g. if the components are not spherical). Note that the constant in the separation depends on w_{min}, and we assume we know this parameter (or a lower bound on it).

The basic idea behind the algorithm is to project the mixture onto $\log k$ dimensions uniformly at random. This projection will preserve distances between each pair of centers μ_i and μ_j with high probability, but will contract distances between samples from the same component and make each component closer to spherical, thus making it easier to cluster. We can then cluster all of the samples into which component generated them, and then for each cluster we can choose the empirical mean and empirical covariance which will with high probability be a good estimate of μ_i and Σ_i. Additionally we can estimate w_i by how large each cluster is.

Informally, we can think of this separation condition as: if we think of each Gaussian as a spherical ball, then if the components are far enough apart then these balls will be *disjoint*.

Arora and Kannan [18], Dasgupta and Schulman [53] – $\widetilde{\Omega}(n^{1/4})$ Separation

We will describe the approach in [18] in detail. The basic question is, if $\sqrt{n}$ separation is the threshold when we can think of the components as disjoint, then how can we learn when the components are much closer? In fact, even if the components are only $\widetilde{\Omega}(n^{1/4})$ separated then it is still true that *every* pair of samples from the same component is closer than *every* pair of samples from different components. How can this be? The explanation is that even though the balls representing each component are no longer disjoint, we are still very unlikely to sample from their overlap region.

Consider $x, x' \leftarrow F_1$ and $y \leftarrow F_2$.

Claim 6.2.3 *All of the vectors $x-\mu_1$, $x'-\mu_1$, $\mu_1-\mu_2$, $y-\mu_2$ are nearly orthogonal (whp)*

This claim is immediate since the vectors $x-\mu_1$, $x'-\mu_1$, $y-\mu_2$ are uniform from a sphere, and $\mu_1-\mu_2$ is the only fixed vector. In fact, any set of vectors in which all but one is uniformly random from a sphere are nearly orthogonal.

Now we can compute:

$$\begin{aligned}\|x-x'\|^2 &\approx \|x-\mu_1\|^2 + \|\mu_1-x'\|^2 \\ &\approx 2n\sigma^2 \pm 2\sigma^2\sqrt{n\log n}\end{aligned}$$

And similarly:

$$\begin{aligned}\|x-y\|^2 &\approx \|x-\mu_1\|^2 + \|\mu_1-\mu_2\|^2 + \|\mu_2-y\|^2 \\ &\approx 2n\sigma^2 + \|\mu_1-\mu_2\|^2 \pm 2\sigma^2\sqrt{n\log n}\end{aligned}$$

Hence if $\|\mu_1-\mu_2\| = \widetilde{\Omega}(n^{1/4},\sigma)$ then $\|\mu_1-\mu_2\|^2$ is larger than the error term and each pair of samples from the same component will be closer than each pair from different components. Indeed we can find the right threshold τ and correctly cluster all of the samples. Again, we can output the empirical mean, empirical covariance and relative size of each cluster and these will be good estimates of the true parameters.

Vempala and Wang [117] – $\widetilde{\Omega}(k^{1/4})$ Separation

Vempala and Wang [117] removed the dependence on n, and replaced it with a separation condition that depends on k – the number of components. The idea is that if we could project the mixture into the subspace T spanned by $\{\mu_1,\dots,\mu_k\}$, we would preserve the separation between each pair of components but reduce the ambient dimension.

So how can we find T, the subspace spanned by the means? We will restrict our discussion to a mixture of spherical Gaussians with a common variance $\sigma^2 I$. Let $x \sim F$ be a random sample from the mixture, then we can write $x = c + z$ where $z \sim N(0,\sigma^2 I_n)$ and c is a random vector that takes the value μ_i with probability w_i for each $i \in [k]$. So:

$$\mathbb{E}[xx^T] = E[cc^T] + E[zz^T] = \sum_{i=1}^{k} w_i\mu_i\mu_i^\top + \sigma^2 I_n$$

Hence the top left singular vectors of $\mathbb{E}[xx^T]$ whose singular value is strictly larger than σ^2 exactly span T. We can then estimate $\mathbb{E}[xx^T]$ from sufficiently many random samples, compute its singular value decomposition and project the mixture onto T and invoke the algorithm of [18].

Brubaker and Vempala [32] – Separating Hyperplane

What if the largest variance of any component is much larger than the separation between the components? Brubaker and Vempala [32] observed that none of the existing algorithms succeed for the *parallel pancakes* example, depicted in Figure **??** even though there is a hyperplane that separates the mixture so that almost all of one component is on one side, and almost all of the other component is on the other side. [32] gave an algorithm that succeeds, provided there is such a separating hyperplane, however the conditions are more complex to state for mixtures of more than two Gaussians. Note that not all mixtures that we could hope to learn have such a separating hyperplane. See e.g. Figure **??**.

6.3 Discussion of Density Estimation

The algorithms we have discussed so far [45], [53], [18], [117], [1], [32] have focused on clustering; can we give efficient learning algorithms even when clustering is *impossible*? Consider a mixture of two Gaussians $F = w_1 F_1 + w_2 F_2$. The separation conditions we have considered so far each imply that $d_{TV}(F_1, F_2) = 1 - o(1)$. In particular, the components have negligible overlap. However if $d_{TV}(F_1, F_2) = 1/2$ we cannot hope to learn which component generated each sample.

More precisely, the total variation distance between two distributions F and G measures how well we can couple them:

Definition 6.3.1 *A coupling between F and G is a distribution on pairs (x, y) so that the marginal distribution on x is F and the marginal distribution on y is G. The error is the probability that $x \neq y$.*

Claim 6.3.2 *There is a coupling with error ε between F and G if and only if $d_{TV}(F, G) \leq \varepsilon$.*

Returning to the problem of clustering the samples from a mixture of two Gaussians, we have that if $d_{TV}(F_1, F_2) = 1/2$ then there is a coupling between F_1 and F_2 that agrees with probability $1/2$. Hence instead of thinking about sampling from a mixture of two Gaussians in the usual way (choose which component, then choose a random sample from it) we can alternatively sample as follows:

(a) Choose (x, y) from the best coupling between F_1 and F_2

(b) If $x = y$, output x

(c) Else output x with probability w_1, and otherwise output y

This procedure generates a random sample from F, but for half of the samples we did not need to decide which component generated it at all! Hence *even if we knew the mixture* there is no clustering procedure that can correctly classify a polynomial number of samples into which component generated them! So in the setting where $d_{TV}(F_1, F_2)$ is not $1 - o(1)$, the fundamental approach we have discussed so far does not work! Nevertheless we will be able to give algorithms to learn the parameters of F even when $d_{TV}(F_1, F_2) = o(1)$ and the components almost entirely overlap.

Next we will discuss some of the basic types of goals for learning algorithms:

(a) **Improper Density Estimation**

Throughout we will assume that $F \in \mathcal{C}$ where $\mathcal{C}$ is some class of distributions (e.g. mixtures of two Gaussians). Our goal in improper density estimation is to find any distribution $\widehat{F}$ so that $d_{TV}(F, \widehat{F}) \leq \varepsilon$. This is the weakest goal for a learning algorithm. A popular approach (especially in low dimension) is to construct a *kernel density estimate*; suppose we take many samples from F and construct a point-mass distribution G that represents our samples. Then we can set $\widehat{F} = G * \mathcal{N}(0, \sigma^2)$, and if F is smooth enough and we take enough samples, $d_{TV}(F, \widehat{F}) \leq \varepsilon$. However $\widehat{F}$ works without learning anything about the components of F; it works just because F is smooth. We remark that such an approach fails badly in high dimensions where even if F is smooth, we would need to take an exponential number of samples in order to guarantee that $\widehat{F} = G * \mathcal{N}(0, \sigma^2 I)$ is close to F.

(b) **Proper Density Estimation**

Here, our goal is to find a distribution $\widehat{F} \in \mathcal{C}$ where $d_{TV}(F, \widehat{F}) \leq \varepsilon$. Note that if $\mathcal{C}$ is the set of mixtures of two Gaussians, then a kernel density estimate is not a valid hypothesis since it will in general be a mixture of many Gaussians (as many samples as we take). Proper density estimation is in general much harder to do than improper density estimation. In fact, we will focus on an even stronger goal:

(b) **Parameter Learning**

Here we require not only that $d_{TV}(F, \widehat{F}) \leq \varepsilon$ and that $\widehat{F} \in \mathcal{C}$, but we want $\widehat{F}$ to be a good estimate for F *on a component-by-component basis*. For example, our goal specialized to the case of mixtures of two Gaussians is:

Definition 6.3.3 *We will say that a mixture $\widehat{F} = \widehat{w}_1\widehat{F}_1 + \widehat{w}_2\widehat{F}_2$ is ε-close (on a component-by-component basis) to F if there is a permutation $\pi : \{1,2\} \to \{1,2\}$ so that for all $i \in \{1,2\}$:*

$$\left|w_i - \widehat{w}_{\pi(i)}\right|, d_{TV}(F_i, \widehat{F}_{\pi(i)}) \leq \varepsilon$$

Note that F and $\widehat{F}$ must necessarily be close as mixtures too: $d_{TV}(F, \widehat{F}) \leq 4\varepsilon$. However we can have mixtures F and $\widehat{F}$ that are both mixtures of k Gaussians, are close as distributions but are not close on a component-by-component basis. It is better to learn F on a component-by-component basis than to do only proper density estimation, if we can. Note that if $\widehat{F}$ is ε-close to F, then even when we cannot cluster samples we will still be able to approximately compute the posterior [79] and this is one of the main advantages of parameter learning over some of the weaker learning goals.

But one should keep in mind that *lower bounds for parameter learning do not imply lower bounds for proper density estimation.* We will give optimal algorithms for parameter learning for mixtures of k Gaussians, which run in polynomial time for any $k = O(1)$. Moreover there are pairs of mixtures of k Gaussians F and $\widehat{F}$ that are not close on a component-by-component basis, but have $d_{TV}(F, \widehat{F}) \leq 2^{-k}$ [95]. Hence there is no algorithm for parameter learning that takes $\mathrm{poly}(n, k, 1/\varepsilon)$ samples – because we need to take at least 2^k samples to distinguish F and $\widehat{F}$. But in the context of proper density estimation, we do not need to distinguish these two mixtures.

Open Question 2 *Is there a* $\mathrm{poly}(n, k, 1/\varepsilon)$ *time algorithm for proper density estimation for mixtures of k Gaussians in n dimensions?*

6.4 Clustering-Free Algorithms

Recall, our goal is to learn $\widehat{F}$ that is ε-close to F. In fact, the same definition can be generalized to mixtures of k Gaussians:

Definition 6.4.1 *We will say that a mixture $\widehat{F} = \sum_{i=1}^{k} \widehat{w}_i\widehat{F}_i$ is ε-close (on a component-by-component basis) to F if there is a permutation $\pi : \{1,2,...,k\} \to \{1,2,...,k\}$ so that for all $i \in \{1,2,...,k\}$:*

$$\left|w_i - \widehat{w}_{\pi(i)}\right|, d_{TV}(F_i, \widehat{F}_{\pi(i)}) \leq \varepsilon$$

When can we hope to learn an ε close estimate in $\text{poly}(n, 1/\varepsilon)$ samples? In fact, there are two trivial cases where we cannot do this, but these will be the only things that go wrong:

(a) If $w_i = 0$, we can never learn $\widehat{F}_i$ that is close to F_i because we never get any samples from F_i.

In fact, we need a quantitative lower bound on each w_i, say $w_i \geq \varepsilon$ so that if we take a reasonable number of samples we will get at least one sample from each component.

(b) If $d_{TV}(F_i, F_j) = 0$ we can never learn w_i or w_j because F_i and F_j entirely overlap.

Again, we need a quantitive lower bound on $d_{TV}(F_i, F_j)$, say $d_{TV}(F_i, F_j) \geq \varepsilon$ for each $i \neq j$ so that if we take a reasonable number of samples we will get at least one sample from the non-overlap region between various pairs of components.

Theorem 6.4.2 *[79], [95] If $w_i \geq \varepsilon$ for each i and $d_{TV}(F_i, F_j) \geq \varepsilon$ for each $i \neq j$, then there is an efficient algorithm that learns an ε-close estimate $\widehat{F}$ to F whose running time and sample complexity are* $\text{poly}(n, 1/\varepsilon, \log 1/\delta)$ *and succeeds with probability* $1 - \delta$.

Note that the degree of the polynomial depends polynomially on k. Kalai, Moitra and Valiant [79] gave the first algorithm for learning mixtures of two Gaussians with no separation conditions. Subsequently Moitra and Valiant [95] gave an algorithm for mixtures of k Gaussians, again with no separation conditions.

In independent and concurrent work, Belkin and Sinha [23] gave a polynomial time algorithm for mixtures of k Gaussians too, however there is no explicit bound given on the running time as a function of k (since their work depends on the basis theorem, which is provably ineffective). Also, the goal in [79] and [95] is to learn $\widehat{F}$ so that its components are close in total variation distance to those of F, which is in general a stronger goal than requiring that the parameters be additively close which is the goal in [23]. The benefit is that the algorithm in [23] works for more general learning problems in the one-dimensional setting, and we will describe this algorithm in detail at the end of this chapter.

Throughout this section, we will focus on the $k = 2$ case since this algorithm is conceptually much simpler. In fact, we will focus on a weaker learning goal: We will say that $\widehat{F}$ is *additively* ε-close to F if $|w_i - \widehat{w}_{\pi(i)}|, \|\mu_i - \widehat{\mu}_{\pi(i)}\|, \|\Sigma_i - \widehat{\Sigma}_{\pi(i)}\|_F \leq \varepsilon$ for all i. We will further assume that F is normalized appropriately:

Definition 6.4.3 *A distribution F is in isotropic position if*

(a) $\mathbb{E}_{x \leftarrow F}[x] = 0$

(b) $\mathbb{E}_{x \leftarrow F}[xx^T] = I$

Alternatively, we require that the mean of the distribution is zero and that its variance *in every direction* is one. In fact this condition is not quite so strong as it sounds:

Claim 6.4.4 *If $\mathbb{E}_{x \leftarrow F}[xx^T]$ is full-rank, then there is an affine transformation that places F in isotropic position*

Proof: Let $\mu = E_{x \leftarrow F}[x]$ and let $E_{x \leftarrow F}[(x-\mu)(x-\mu)^T] = M$. It is easy to see that M is positive semi-definite, and in fact is full rank by assumption. Hence we can write $M = BB^T$ where B is invertible (this is often referred to as the Cholesky decomposition [74]). Then set $y = B^{-1}(x - \mu)$, and it is easy to see that $\mathbb{E}[y] = 0$ and $\mathbb{E}[yy^T] = B^{-1}M(B^{-1})^T = I$. ■

Our goal is to learn an additive ε approximation to F, and we will assume that F has been pre-processed so that it is in isotropic position.

Outline

We can now describe the basic outline of the algorithm, although there will be many details to fill:

(a) Consider a series of projections down to one dimension

(b) Run a univariate learning algorithm

(c) Set up a system of linear equations on the high-dimensional parameters, and back solve

Isotropic Projection Lemma

We will need to overcome a number of obstacles to realize this plan, but let us work through the details of this outline:

Claim 6.4.5 $proj_r[\mathcal{N}(\mu, \Sigma)] = \mathcal{N}(r^T\mu, r^T\Sigma r)$

Alternatively, the projection of a high-dimensional Gaussian is a one-dimensional Gaussian, and its mean and variance are $r^T\mu$ and $r^T\Sigma r$ respectively. This implies that if we knew the parameters of the projection of a single Gaussian component onto a (known) direction r, then we could use these parameters to set up a linear constraint for μ and Σ. If we follow this plan, we would need to consider about n^2 projections to get enough linear constraints, since there are $\Theta(n^2)$ variances in Σ that we need to solve for. Now we will encounter the first problem in the outline. Let us define some notation:

Definition 6.4.6 $d_p(\mathcal{N}(\mu_1,\sigma_1^2),\mathcal{N}(\mu_2,\sigma_2^2)) = |\mu_1-\mu_2| + |\sigma_1^2-\sigma_2^2|$

We will refer to this as the parameter distance. Ultimately, we will give a univariate algorithm for learning mixtures of Gaussians and we would like to run it on $\text{proj}_r[F]$.

Problem 4 *But what if $d_p(proj_r[F_1], proj_r[F_2])$ is exponentially small?*

This would be a problem since we would need to run our univariate algorithm with exponentially fine precision just to see that there are two components and not one! How can we get around this issue? In fact, this almost surely never happens provided that F is in isotropic position. For intuition, consider two cases:

(a) Suppose $\|\mu_1-\mu_2\| \geq \text{poly}(1/n,\varepsilon)$.

If the difference between the means of F_1 and F_2 is at least any fixed inverse polynomial, then with high probability $\|r^T\mu_1 - r^T\mu_2\|$ is at least $\text{poly}(1/n,\varepsilon)$ too. Hence $\text{proj}_r[F_1]$ and $\text{proj}_r[F_2]$ will have different parameters due to a difference in their means.

(b) Suppose $\|\mu_1-\mu_2\| \leq \text{poly}(1/n,\varepsilon)$.

The key observation is that if $d_{TV}(F_1,F_2)\geq\varepsilon$ and their means are almost identical, then their covariances Σ_1 and Σ_2 must be noticeably different when projected on a random direction r. In this case, $\text{proj}_r[F_1]$ and $\text{proj}_r[F_2]$ will have different parameters due to a difference in their variances. This is the intuition behind the following lemma:

Lemma 6.4.7 *If F is in isotropic position and $w_i \geq \varepsilon$ and $d_{TV}(F_1,F_2)\geq\varepsilon$, then with high probability for a random r*

$$d_p(proj_r[F_1], proj_r[F_2]) \geq 2\varepsilon_3 = \text{poly}(1/n,\varepsilon)$$

Note that this lemma is note true when F is not in isotropic position (e.g. consider the parallel pancakes example), and moreover when generalizing to mixtures of $k > 2$ Gaussians this is the key step that fails since even if F is in isotropic position, it could be that for almost all choices of r the projection onto r results in a mixtures that is exponentially closet to a mixture of $< k$ Gaussians! (The approach in [95] is to learn a mixture of $< k$ Gaussians as a proxy for the true mixture, and later on find a direction that can be used to cluster the mixture into sub mixtures and recurse).

Pairing Lemma

Next we will encounter the second problem: Suppose we project onto direction r and s and learn $\widehat{F}^r = \frac{1}{2}\widehat{F}^r_1 + \frac{1}{2}\widehat{F}^r_2$ and $\widehat{F}^s = \frac{1}{2}\widehat{F}^s_1 + \frac{1}{2}\widehat{F}^s_2$ respectively. Then the mean and variance of $\widehat{F}^r_1$ yield a linear constraint on one of the two high-dimensional Gaussians, and similarly for $\widehat{F}^s_1$.

Problem 5 *How do we know that they yield constraints on the* same *high-dimensional component?*

Ultimately we want to set up a system of linear constraints to solve for the parameters of F_1, but when we project F onto different directions (say, r and s) we need to pair up the components from these two directions. The key observation is that as we vary r to s the parameters of the mixture vary continuously. See Figure **??**. Hence when we project onto r, we know from the isotropic projection lemma that the two components will either have noticeably different means or variances. Suppose their means are different by ε_3; then if r and s are close (compared to ε_1) the parameters of each component in the mixture do not change much and the component in $\text{proj}_r[F]$ with larger mean will correspond to the same component as the one in $\text{proj}_s[F]$ with larger mean. A similar statement applies when it is the variances that are at least ε_3 apart.

Lemma 6.4.8 *If* $\|r - s\| \leq \varepsilon_2 = \text{poly}(1/n, \varepsilon_3)$ *then*

(a) If $|r^T\mu_1 - r^T\mu_2| \geq \varepsilon_3$ then the components in $proj_r[F]$ and $proj_s[F]$ with the larger mean correspond to the same high-dimensional component

(b) Else if $|r^T\Sigma_1 r - r^T\Sigma_2 r| \geq \varepsilon_3$ then the components in $proj_r[F]$ and $proj_s[F]$ with the larger variance correspond to the same high-dimensional component

Hence if we choose r randomly and only search over directions s with $\|r - s\| \leq \varepsilon_2$, we will be able to pair up the components correctly in the different one-dimensional mixtures.

Condition Number Lemma

Now we encounter the final problem in the high-dimensional case: Suppose we choose r randomly and for $s_1, s_2, ..., s_p$ we learn the parameters of the projection of F onto these directions and pair up the components correctly. We can only hope to learn the parameters on these projection up to some additive accuracy ε_1 (and our univariate learning algorithm will have running time and sample complexity $\text{poly}(1/\varepsilon_1)$).

Problem 6 *How do these errors in our univariate estimates translate to errors in our high dimensional estimates for $\mu_1, \Sigma_1, \mu_2, \Sigma_2$?*

Recall that the *condition number* controls this. The final lemma we need in the high-dimensional case is:

Lemma 6.4.9 *The condition number of the linear system to solve for μ_1, Σ_1 is* $\text{poly}(1/\varepsilon_2, n)$ *where all pairs of directions are ε_2 apart.*

Intuitively, as r and $s_1, s_2, ..., s_p$ are closer together then the condition number of the system will be worse (because the linear constraints are closer to redundant), but the key fact is that the condition number is bounded by a fixed polynomial in $1/\varepsilon_2$ and n, and hence if we choose $\varepsilon_1 = \text{poly}(\varepsilon_2, n)\varepsilon$ then our estimates to the high-dimensional parameters will be within an additive ε. Note that each parameter $\varepsilon, \varepsilon_3, \varepsilon_2, \varepsilon_1$ is a fixed polynomial in the earlier parameters (and $1/n$) and hence we need only run our univariate learning algorithm with inverse polynomial precision on a polynomial number of mixtures to learn an ε-close estimate $\widehat{F}$!

But we still need to design a univariate algorithm, and next we return to Pearson's original problem!

6.5 A Univariate Algorithm

Here we will give a univariate algorithm to learning the parameters of a mixture of two Gaussians up to additive accuracy ε whose running time and sample complexity is $\text{poly}(1/\varepsilon)$. Note that the mixture $F = w_1F_1 + w_2F_2$ is in isotropic position (since the projection of a distribution in isotropic position is itself in isotropic position), and as before we assume $w_1, w_2 \geq \varepsilon$ and $d_{TV}(F_1, F_2) \geq \varepsilon$. Our first observation is that all of the parameters are bounded:

Claim 6.5.1 *(a)* $\mu_1, \mu_2 \in [-1/\sqrt{\varepsilon}, 1/\sqrt{\varepsilon}]$

(b) $\sigma_1^2, \sigma_2^2 \in [0, 1/\varepsilon]$

This claim is immediate, since if any of the above conditions are violated it would imply that the mixture has variance strictly larger than one (because $w_1, w_2 \geq \varepsilon$ and the mean of the mixture is zero).

Hence we could try to learn the parameters using a *grid search*:

```
Grid Search
Input: samples from $F(\Theta)$
Output: parameters $\widehat{\Theta} = (\widehat{w}_1, \widehat{\mu}_1, \widehat{\sigma}_1^2, \widehat{\mu}_2, \widehat{\sigma}_2^2)$

For all valid $\widehat{\Theta}$ where the parameters are multiples of $\varepsilon^C$
   Test $\widehat{\Theta}$ using the samples, if it passes output $\widehat{\Theta}$
End
```

For example, we could test out $\widehat{\Theta}$ by computing the first six moments of $F(\Theta)$ from enough random examples, and output $\widehat{\Theta}$ if its first six moments are each within an additive τ of the observed moments. (This is a slight variant on Pearson's sixth moment test).

It is easy to see that if we take enough samples and set τ appropriately, then if we round the true parameters Θ to any valid grid point whose parameters are multiples of ε^C, then the resulting $\widehat{\Theta}$ will with high probability pass our test. This is called the *completeness*. The much more challenging part is establishing the *soundness*; after all why is there no other set of parameters $\widehat{\Theta}$ except for ones close to Θ that pass our test?

Alternatively, we want to prove that any two mixtures F and $\widehat{F}$ whose parameters *do not* match within an additive ε must have one of their first six moments noticeably different. The main lemma is:

Lemma 6.5.2 *For any F and $\widehat{F}$ that are not ε-close in parameters, there is an $r \in \{1, 2, ..., 6\}$ where*

$$\left| M_r(\Theta) - M_r(\widehat{\Theta}) \right| \geq \varepsilon^{O(1)}$$

where Θ and $\widehat{\Theta}$ are the parameters of F and $\widehat{F}$ respectively, and M_r is the r^{th} raw moment.

Let $\widetilde{M_r}$ be the empirical moments. Then

$$\left| M_r(\widehat{\Theta}) - M_r(\Theta) \right| \leq \underbrace{\left| \widetilde{M_r}(\widehat{\Theta}) - \widetilde{M_r} \right|}_{\leq \tau} + \underbrace{\left| \widetilde{M_r} - M_r(\Theta) \right|}_{\leq \tau} \leq 2\tau$$

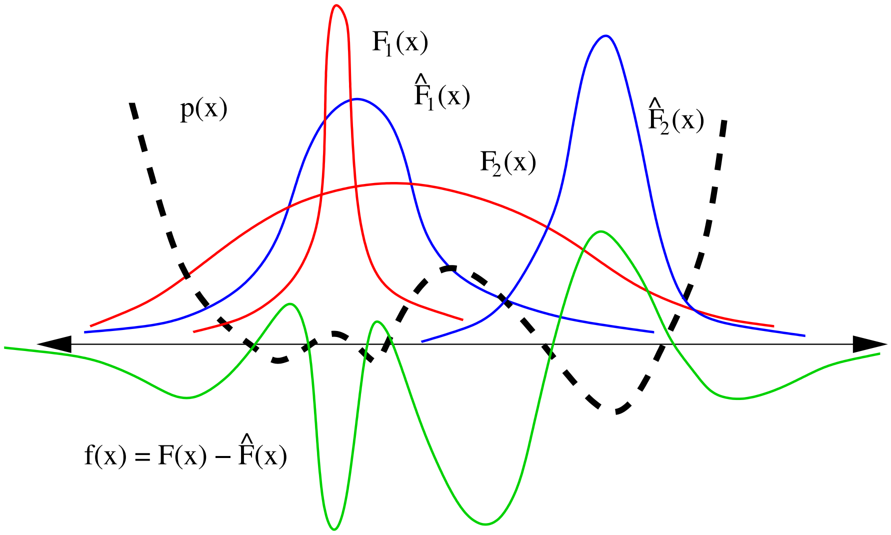

Figure 6.2: If $f(x)$ has at most six zero crossings, we can find at most degree six polynomial that agrees with its sign

where the first term is at most τ because the test passes and the second term is small because we can take enough samples (but still poly$(1/\tau)$) so that the empirical moments and the true moments are close. Hence we can apply the above lemma in the contrapositive, and conclude that if the grid search outputs $\widehat{\Theta}$ then Θ and $\widehat{\Theta}$ must be ε-close in parameters, which gives us an efficient univariate algorithm!

So our main goal is to prove that if F and $\widehat{F}$ that are not ε-close, then one of their first six moments is noticeably different. In fact, even the case of $\varepsilon = 0$ is challenging: If F and $\widehat{F}$ are different mixtures of two Gaussians, why is one of their first six moments necessarily different? Our main goal is to prove this statement, using the *heat equation*.

In fact, let us consider the following thought experiment. Let $f(x) = F(x) - \widehat{F}(x)$ be the point-wise difference between the density functions F and $\widehat{F}$. Then, the heart of the problem is: Can we prove that $f(x)$ crosses the x-axis at most six times? See Figure 6.2.

Lemma 6.5.3 *If $f(x)$ crosses the x-axis at most six times, then one of the first six moments of F and $\widehat{F}$ are different*

Proof: In fact, we can construct a (non-zero) degree at most six polynomial $p(x)$ that agrees with the sign of $f(x)$ – i.e. $p(x)f(x) \geq 0$ for all x. Then

$$\begin{aligned} 0 < \left| \int_x p(x) f(x) dx \right| &= \left| \int_x \sum_{r=1}^{6} p_r x^r f(x) dx \right| \\ &\leq \sum_{r=1}^{6} |p_r| \left| M_r(\Theta) - M_r(\widehat{\Theta}) \right| \end{aligned}$$

And if the first six moments of F and $\widehat{F}$ match exactly, the right hand side is zero which is a contradiction. ■

So all we need to prove is that $F(x) - \widehat{F}(x)$ has at most six zero crossings. Let us prove a stronger lemma by induction:

Lemma 6.5.4 *Let $f(x) = \sum_{i=1}^{k} \alpha_i \mathcal{N}(\mu_i, \sigma_i^2, x)$ be a linear combination of k Gaussians (α_i can be negative). Then if $f(x)$ is not identically zero, $f(x)$ has at most $2k - 2$ zero crossings.*

We will rely on the following tools:

Theorem 6.5.5 *Given $f(x) : \mathbb{R} \to \mathbb{R}$, that is analytic and has n zero crossings, then for any $\sigma^2 > 0$, the function $g(x) = f(x) * \mathcal{N}(0, \sigma^2)$ has at most n zero crossings.*

This theorem has a physical interpretation. If we think of $f(x)$ as the heat profile of an infinite one-dimensional rod, then what does the heat profile look like at some later time? In fact it is precisely $g(x) = f(x) * \mathcal{N}(0, \sigma^2)$ for an appropriately chosen σ^2. Alternatively, the Gaussian is the *Green's function* of the heat equation. And hence many of our physical intuitions for diffusion have consequences for convolution – convolving a function by a Gaussian has the effect of smoothing it, and it cannot create a new local maxima (and relatedly it cannot create new zero crossings).

Finally we recall the elementary fact:

Fact 6.5.6 $\mathcal{N}(0, \sigma_1^2) * \mathcal{N}(0, \sigma_2^2) = \mathcal{N}(0, \sigma_1^2 + \sigma_2^2)$

Now we are ready to prove the above lemma and conclude that if we knew the first six moments of a mixture of two Gaussians *exactly*, then we would know its parameters exactly too. Let us prove the above lemma by induction, and assume that for any linear combination of $k = 3$ Gaussians, the number of zero crossings is

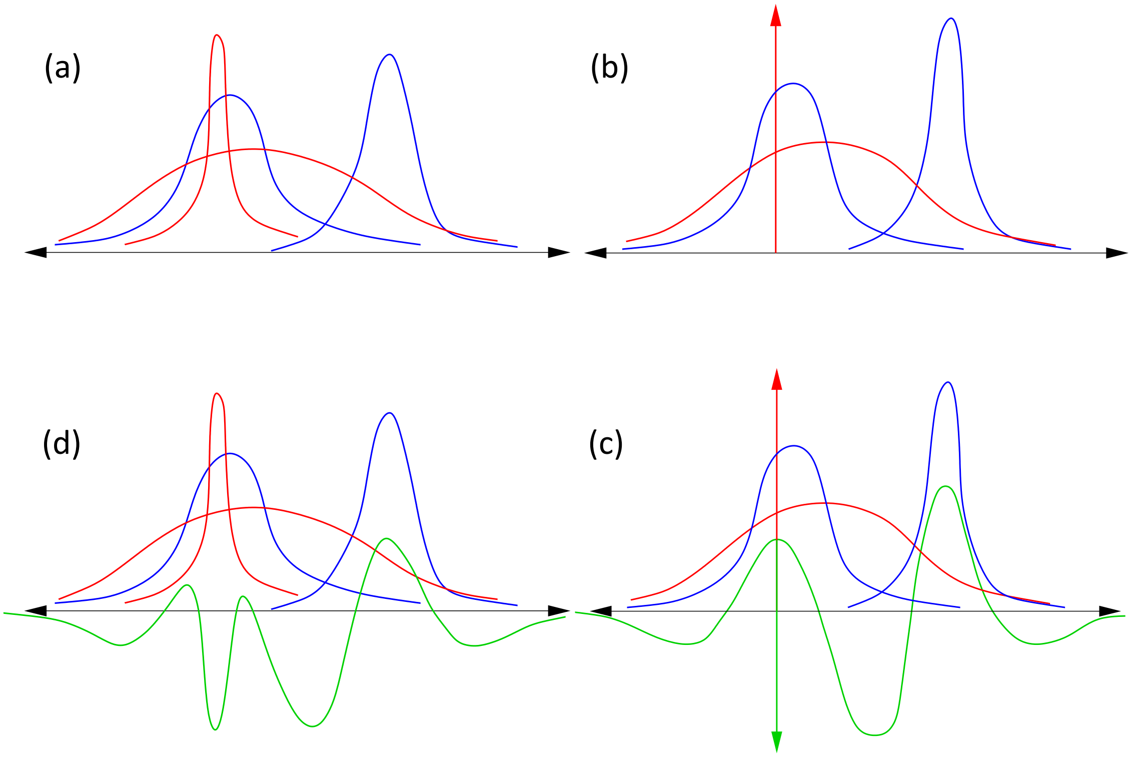

Figure 6.3: (a) linear combination of four Gaussians (b) subtracting σ^2 from each variance (c) adding back in the delta function (d) convolving by $\mathcal{N}(0, \sigma^2)$ to recover the original linear combination

at most four. Now consider an arbitrary linear combination of four Gaussians, and let σ^2 be the smallest variance of any component. See Figure 6.3(a). We can consider a related mixture where we subtract σ^2 from the variance of each component. See Figure 6.3(b).

Now if we ignore the delta function, we have a linear combination of three Gaussians and by induction we know that it has at most four zero crossings. But how many zero crossings can we add when we add back in the delta function? We can add at most two, one on the way up and one on the way down (here we are ignoring some real analysis complications of working with delta functions for ease of presentation). See Figure 6.3(c). And now we can convolve the function by $\mathcal{N}(0, \sigma^2)$ to recover the original linear combination of four Gaussians, but this last step does not increase the number of zero crossings! See Figure 6.3(d).

This proves that

$$\left\{M_r(\widehat{\Theta}) = M_r(\Theta)\right\},\ r = 1, 2, ..., 6$$

has only two solutions (the true parameters and we can also interchange which is component is which). In fact, this system of polynomial equations is also *stable* and there is an analogue of condition numbers for systems of polynomial equations that implies a quantitative version of what we have just proved: if F and $\widehat{F}$ that are not ε-close, then one of their first six moments is noticeably different. This gives us our univariate algorithm.

6.6 A View from Algebraic Geometry

Here we will present an alternative univariate learning algorithm of Belkin and Sinha [23] that also makes use of the method of moments, but gives a much more general analysis using tools from algebraic geometry.

Polynomial Families

We will analyze the method of moments for the following class of distributions:

Definition 6.6.1 *A class of distributions $F(\Theta)$ is called a polynomial family if*

$$\forall r,\ \mathbb{E}_{X \in F(\Theta)}\left[X^r\right] = M_r(\Theta)$$

where $M_r(\Theta)$ is a polynomial in $\Theta = (\theta_1, \theta_2, ..., \theta_k)$.

This definition captures a broad class of distributions such as mixtures models whose components are uniform, exponential, Poisson, Gaussian or gamma functions. We will need another (tame) condition on the distribution which guarantees that it is characterized by all of its moments.

Fact 6.6.2 *If the moment generating function (mgf) of X defined as $\sum \mathbb{E}[X^n] \frac{t^n}{n!}$ converges in a neighborhood of zero, it uniquely determines the probability distribution, i.e.*

$$\forall r,\ M_r(\Theta) = M_r(\widehat{\Theta}) \implies F(\Theta) = F(\widehat{\Theta}).$$

Our goal is to show that for any polynomial family, a *finite* number of its moments suffice. First we introduce the relevant definitions:

Definition 6.6.3 *Given a ring R, an ideal I generated by $g_1, g_2, \cdots, g_n \in R$ denoted by $I = \langle g_1, g_2, \cdots, g_n \rangle$ is defined as*

$$I = \left\{ \sum_i r_i g_i \textit{ where } r_i \in R \right\}.$$

Definition 6.6.4 *A Noetherian ring is a ring such that for any sequence of ideals*

$$I_1 \subseteq I_2 \subseteq I_3 \subseteq \cdots,$$

there is N such that $I_N = I_{N+1} = I_{N+2} = \cdots$.

Theorem 6.6.5 (Hilbert's Basis Theorem) *If R is a Noetherian ring, then $R[X]$ is also a Noetherian ring.*

It is easy to see that $\mathbb{R}$ is a Noetherian ring, and hence we know that $\mathbb{R}[x]$ is also Noetherian. Now we can prove that for any polynomial family, a finite number of moments suffice to uniquely identify any distribution in the family:

Theorem 6.6.6 *Let $F(\Theta)$ be a polynomial family. If the moment generating function converges in a neighborhood of zero, there exists N such that*

$$F(\Theta) = F(\widehat{\Theta}) \textit{ if and only if } M_r(\Theta) = M_r(\widehat{\Theta})\ \forall r \in 1, 2, \cdots, N$$

Proof: Let $Q_r(\Theta, \widehat{\Theta}) = M_r(\Theta) - M_r(\widehat{\Theta})$. Let $I_1 = \langle Q_1 \rangle$, $I_2 = \langle Q_1, Q_2 \rangle, \cdots$. This is our ascending chain of ideals in $\mathbb{R}[\Theta, \widehat{\Theta}]$. We can invoke Hilbert's basis theorem and conclude that $\mathbb{R}[X]$ is a Noetherian ring and hence, there is N such that $I_N = I_{N+1} = \cdots$. So for all $N + j$, we have

$$Q_{N+j}(\Theta, \widehat{\Theta}) = \sum_{i=1}^{N} p_{ij}(\Theta, \widehat{\Theta}) Q_i(\Theta, \widehat{\Theta})$$

for some polynomial $p_{ij} \in \mathbb{R}[\Theta, \widehat{\Theta}]$. Thus, if $M_r(\Theta) = M_r(\widehat{\Theta})$ for all $r \in 1, 2, \cdots, N$, then $M_r(\Theta) = M_r(\widehat{\Theta})$ for all r and from Fact 6.6.2 we conclude that $F(\Theta) = F(\widehat{\Theta})$.

The other side of the theorem is obvious. ■

The theorem above does not give any finite bound on N, since the basis theorem does not either. This is because the basis theorem is proved by contradiction, but more fundamentally it is not possible to give a bound on N that depends only on the choice of the ring. Consider the following example

Example 1 *Consider the Noetherian ring $\mathbb{R}[x]$. Let $I_i = \langle x^{N-i} \rangle$ for $i = 0, \cdots, N$. It is a strictly ascending chain of ideals for $i = 0, \cdots, N$. Therefore, even if the ring $\mathbb{R}[x]$ is fixed, there is no universal bound on N.*

Bounds such as those in Theorem 6.6.6 are often referred to as *ineffective*. Consider an application of the above result to mixtures of Gaussians: from the above theorem, we have that any two mixtures F and $\widehat{F}$ of k Gaussians are identical if and only if these mixtures agree on their first N moments. Here N is a function of k, and N is finite but we cannot write down any explicit bound on N as a function of k using the above tools. Nevertheless, these tools apply much more broadly than the specialized ones based on the heat equation that we used to prove that $4k - 2$ moments suffice for mixtures of k Gaussians in the previous section.

Systems of Polynomial Inequalities

In general, we do not have exact access to the moments of a distribution but only noisy approximations. Our main goal is to prove a quantitive version of the previous result which shows that any two distributions F and $\widehat{F}$ that are close on their first N moments are close in their parameters too. The key fact is that we can bound the condition number of systems of polynomial inequalities; there are a number of ways to do this but we will use *quantifier elimination*. Recall:

Definition 6.6.7 *A set S is semi-algebraic if there exist multivariate polynomials $p_1, ..., p_n$ such that*

$$S = \{x_1, ..., x_r | p_i(x_1, ..., x_r) \geq 0\}$$

or if S is a finite union or intersection of such sets.

Theorem 6.6.8 (Tarski) *The projection of a semi-algebraic set is semi-algebraic.*

We define the following helper set:

$$H(\varepsilon, \delta) = \Big\{\forall(\Theta, \widehat{\Theta}) \ : \ |M_r(\Theta) - M_r(\widehat{\Theta})| \leq \delta \text{ for } r = 1, 2, ...N \Longrightarrow \|\Theta - \widehat{\Theta}\| \leq \varepsilon\Big\}.$$

Let $\varepsilon(\delta)$ be the smallest ε as a function of δ:

Theorem 6.6.9 *There are fixed constants C_1, C_2, s such that if $\delta < 1/C_1$ then $\varepsilon(\delta) < C_2\delta^{1/s}$.*

Proof: It is easy to see that we can define $H(\varepsilon, \delta)$ as the projection of a semi-algebraic set, and hence using Tarski's theorem we conclude that $H(\varepsilon, \delta)$ is also semi-algebraic. The crucial observation is that because $H(\varepsilon, \delta)$ is semi-algebraic, the smallest that we can choose ε to be as a function of δ is itself a polynomial function of δ. There are some caveats here, because we need to prove that for a fixed δ we can choose ε to be strictly greater than zero and moreover the polynomial relationship between ε and δ only holds if δ is sufficiently small. However these technical issues can be resolved without much more work, see [23] and the main result is the following. ■

Corollary 6.6.10 *If $|M_r(\Theta) - M_r(\widehat{\Theta})| \leq \left(\frac{\varepsilon}{C_2}\right)^s$ then $|\Theta - \widehat{\Theta}| \leq \varepsilon$.*

Hence there is a polynomial time algorithm to learn the parameters of any univariate polynomial family (whose mgf converges in a neighborhood of zero) within an additive accuracy of ε whose running time and sample complexity is poly$(1/\varepsilon)$; we can take enough samples to estimate the first N moments within ε^s and search over a grid of the parameters, and any set of parameters that matches each of the moments is necessarily close in parameter distance to the true parameters.