18.409 Final Project: More General Mixture Models

Wei Hu
huwei@mit.edu

Ariel Schvartzman
arielsc@mit.edu

May 13, 2015

1 Introduction

In this short paper we present techniques for learning a mixture of distributions for more general classes of distributions than the ones seen in class. We focus on two papers, their key theorems and how they are applied in algorithms with provable guarantees.

The first paper, by Kannan et al. [2], uses a spectral projection technique to learn a mixture of log-concave distributions efficiently provided that the means are separated. Their main theorem, which we present and prove, says that for an arbitrary mixture of distributions the SVD subspace of a sample is close to the means of the samples, where the closeness depends on the variances. Their algorithm works for a mixture of log-concave distributions, which generalizes previous results for Gaussians.

The second paper, by Dasgupta et al. [1], focuses on efficiently learning a mixture of symmetric heavy-tailed distributions whose expectation or variance might be infinite, with minimal separation requirements. They present an algorithm for learning when the centers of the distributions are known which uses the ℓ_1 norm as a classifier, assuming some additional restrictions on the distributions. They also present an algorithm that works when the centers are not known and provably works for the family of distributions they consider. We will focus on the first result as it is easier to understand and provides some intuition on the second result.

In both papers, the authors examine different ways to solve the following problem. There is a mixture of k distributions $F_1, \cdots, F_k$ in n dimensions, and the distribution F_i has mixing weight w_i. A sample x from the mixture is taken from distribution F_i with probability w_i. The goal is to classify i.i.d. samples from F and to approximately learn the underlying distributions.

2 Spectral Projection Method

The method of spectral projection, i.e., projecting the samples onto the subspace spanned by the top k singular vectors (the SVD subspace) of the distribution, was successfully applied to the special case of learning a mixture of spherical Gaussians, provided that the separation condition $|\mu_i - \mu_j| = (\sigma_i + \sigma_j)\Omega^*(k^{1/4})$ holds for any $i \neq j$ [3]. This removed the dependence on n in the separation condition. The key idea is that the SVD subspace of a mixture of spherical Gaussians contains the means of its k components. However, this property is not valid for more general classes of distributions, e.g. non-spherical Gaussians. Nevertheless, in [2] Kannan et al. show that spectral projection can approximately preserve the intermean distances "on average". Based on this key result, they give an efficient algorithm for log-concave distributions, under the separation condition $|\mu_i - \mu_j| = (\sigma_i + \sigma_j)\Omega^*\left(\frac{k^{3/2}}{\varepsilon^2}\right)$, where ε is a lower bound on the mixing weights w_i.

It is worth noting that this algorithm for log-concave distributions does not identify all the components after one spectral projection (as in the case of spherical Gaussians). Instead, it runs iteratively and identifies one component in each iteration through projection. The reason is that one can only show that the intermean distances are preserved in an average sence after projection, and some component means may become very close so the samples from those components are indistinguishable. On the other hand, it is shown that there is one component that is still separated from others, and the algorithm correctly identifies such a component in each iteration.

Notation. We use $|\cdot|$ to denote the ℓ_2 distance in $\mathbb{R}^n$. A mixture F in $\mathbb{R}^n$ has k components $F_1, \cdots, F_k$ with mixing weights $w_1, \cdots, w_k$. The mean of F_i is μ_i, and the maximum variance of F_i in any direction is denoted by σ_i^2. For any subspace W, we denote the maximum variance of F_i along any direction in W by $\sigma_{i,W}^2$. The orthogonal distance from a point x to a subspace W is denoted by $d(x, W)$.

For a set of i.i.d. samples S from F, we partition S as $S = S_1 \cup \cdots \cup S_k$, where S_i is the set of samples from F_i. Let μ_i^S be the sample mean of S_i, i.e., $\mu_i^S = \frac{1}{|S_i|}\sum_{x \in S_i} x$. For any subspace W, we denote the maximum sample variance of S_i along any direction in W by $\hat{\sigma}_{i,W}^2(S)$.

2.1 Main theorem on spectral projection

For a set of points S in $\mathbb{R}^n$, let A be the matrix whose rows are the points in S. Then the subspace spanned by the top k right singular vectors of A is called the *SVD subspace* of S. The following theorem from [2] is an important property of spectral projection. We follow the notations introduced before.

Theorem 1. *Let W be the SVD subspace of S. Then*

$$\sum_{i=1}^{k} |S_i| d(\mu_i^S, W)^2 \leq k \sum_{i=1}^{k} |S_i| \hat{\sigma}_{i,W}^2(S).$$

Proof. We use the following lemma which is easy to verify.

Lemma 1. *For points $p, p_1, \cdots, p_K \in \mathbb{R}^n$, if $\mu_P = \frac{1}{K}\sum_{i=1}^{K} p_i$, then*

$$\sum_{i=1}^{K} |p_i - p|^2 = K|p - \mu_P|^2 + \sum_{i=1}^{K} |p_i - \mu_P|^2.$$

Let M be the span of $\mu_1^S, \cdots, \mu_k^S$. For any $x \in \mathbb{R}^n$, denote by $\pi_M(x)$ the projection of x onto M and by $\pi_W(x)$ the projection of x onto W. For any i, since $\mu_i^S \in M$, from Lemma 1 we have

$$\sum_{x \in S_i} |\pi_M(x)|^2 = |S_i| \cdot |\mu_i^S|^2 + \sum_{x \in S_i} |\pi_M(x) - \mu_i^S|^2 \geq |S_i| \cdot |\mu_i^S|^2.$$

Taking the sum for $i = 1, \cdots, k$, we have

$$\sum_{x \in S} |\pi_M(x)|^2 = \sum_{i=1}^{k} \sum_{x \in S_i} |\pi_M(x)|^2 \geq \sum_{i=1}^{k} |S_i| \cdot |\mu_i^S|^2 = \sum_{i=1}^{k} |S_i| \left(|\pi_W(\mu_i^S)|^2 + d(\mu_i^S, W)^2 \right). \quad (1)$$

Let $e_1, \cdots, e_k$ be an orthonormal basis for W. For any i, using Lemma 1 we have

$$\begin{aligned}\sum_{x\in S_i} |\pi_W(x)|^2 &= |S_i| \cdot |\pi_W(\mu_i^S)|^2 + \sum_{x\in S_i} |\pi_W(x-\mu_i^S)|^2 \\ &= |S_i| \cdot |\pi_W(\mu_i^S)|^2 + \sum_{j=1}^{k}\sum_{x\in S_i} |\pi_W(x-\mu_i^S)\cdot e_j|^2 \\ &\le |S_i| \cdot |\pi_W(\mu_i^S)|^2 + k|S_i|\hat{\sigma}_{i,W}^2(S),\end{aligned}$$

where the last inequality is because the variance of S_i along any direction in W is at most $\hat{\sigma}_{i,W}^2(S)$ (by definition of $\hat{\sigma}_{i,W}^2(S)$). Taking the sum for $i = 1, \cdots, k$, we get

$$\sum_{x\in S} |\pi_W(x)|^2 \le \sum_{i=1}^{k} |S_i| \cdot |\pi_W(\mu_i^S)|^2 + k\sum_{i=1}^{k} |S_i|\hat{\sigma}_{i,W}^2(S) \tag{2}$$

It is well-known that the SVD subspace, among all subspaces of dimension at most k, minimizes the sum of squared distances from points in S to the subspace. Equivalently, it maximizes the sum of squared lengths of projections. Hence $\sum_{x\in S} |\pi_M(x)|^2 \le \sum_{x\in S} |\pi_W(x)|^2$. Then by comparing the RHSs of (1) and (2), the proof is completed. □

Theorem 1 essentially gives a way to lower bound the distances between component means after projection, in an average sense. If the means are well separated, at least some of them will continue to be separated after projection.

2.2 Algorithm by spectral projection

Now we describe the algorithm for learning a mixture of log-concave distributions. The idea is to project the samples onto the SVD subspace and to classify samples in that subspace. However, since Theorem 1 only indicates that the intermean distances are preserved in an average sense, we should not expect to classify all samples after one projection. To overcome this, the algorithm runs in k iterations, with the guarantee that in each iteration there exists one "large" component that is well separated from others. The algorithm identifies this component, deletes the samples coming from it, and continues to the next iteration.

The inputs consist of N i.i.d. samples, a weight lower bound $0 < \varepsilon < 1$, an error probability bound $0 < \delta < 1$, and a parameter $N_0 < N$. For simplicity, we only describe the first iteration of the algorithm: (the rest iterations are essentially the same)

1. Choose a subset of samples S of size N_0. Find the k-dimensional SVD subspace W of S.

2. Delete S and project the remaining samples, T, to the subspace W.

3. For each projected point p:

    - Find its closest $\varepsilon N/2$ points. Let the set of these points be $T(p)$ and their mean be $\mu(p)$.
    - Let $A(p)$ be the matrix whose rows are $x - \mu(p)$ for all $x \in T(p)$. Compute the largest singular value $\sigma(p)$ of $A(p)$. (Note that $\sigma(p)^2$ is the maximum variance of $T(p)$ along any direction in W.)

4. Find a point p_0 which maximizes $\sigma(p_0)$. Let T_0 be the set of all points in T whose projections are within distance $\frac{256\sqrt{k}\log(Nk/\delta)}{\varepsilon}\sigma(p_0)$ from p_0.

5. Identify T_0 as a component and delete it from samples.

It is shown in [2] that this iterative spectral projection algorithm can correctly classify $N - kN_0$ samples with probability at least $1-\delta$ if (i) N_0 is large enough (polynomial in n, $\log k$, $\frac{1}{\varepsilon}$ and $\log \frac{1}{\delta}$); (ii) $N > C\frac{kN_0}{\varepsilon}$ for some universal constant C; (iii) the means of the components are separated as $|\mu_i - \mu_j| \geq 2^{11}(\sigma_i + \sigma_j) \cdot \frac{k^{3/2}}{\varepsilon^2} \cdot \log^2 \frac{Nk}{\delta}$. Note that kN_0 samples are used for computing SVD subspaces and are not classified.

2.3 Sketch of the analysis

We give a sketch of the analysis of the algorithm.

Sample properties. For a log-concave distribution, the distance from a random point to its mean has an exponentially decreasing distribution, so a sufficiently large number of i.i.d. samples will have good concentration properties. For the algorithm, it is shown that if T is a set of at least N_0 samples, then w.h.p. for each i,

(a) $w_i - \frac{\varepsilon}{4} \leq \frac{|T_i|}{|T|} \leq w_i + \frac{\varepsilon}{4}$.

(b) $|\mu_i - \mu_i^T| \leq \frac{\sigma_i}{4}$.

(c) For any subspace W, $\frac{7}{8}\sigma_{i,W}^2 \leq \hat{\sigma}_{i,W}^2(T) \leq \frac{8}{7}\sigma_{i,W}^2$.

Here (a) directly follows from Chernoff bound, which ensures that the number of samples from each component is neither too small nor too large. (b) and (c) state that for each component the mean and variances of samples are close to their true values; they rely on the assumption that all the components are log-concave. In the entire analysis, it is assumed that these properties always hold for the samples.

Variation of Theorem 1. As a technical issue regarding independence, the algorithm has to use a set S of samples to calculate the SVD subspace W and project other samples T onto this subspace, so Theorem 1 cannot be applied directly. However, given the sample properties (a)-(c), an alternative bound can be proved based on Theorem 1:

$$\sum_{i=1}^{k} |T_i| d(\mu_i^S, W)^2 \leq 2k \sum_{i=1}^{k} |T_i| \hat{\sigma}_{i,W}^2(T). \tag{3}$$

Large components. The key step is to show that w.h.p. the algorithm can correctly identify one *large component* in each iteration. A component F_r is large if it satisfies $|T_r|\hat{\sigma}_{r,W}^2(T) \geq \beta \max_i |T_i|\hat{\sigma}_{i,W}^2(T)$, where $\beta = \frac{\varepsilon^3}{2^{14} k \log^2(\frac{Nk}{\delta})}$. Then it can be shown from (3) and the sample properties that after projected onto W, the mean of a large component remains separated from the means of all other components. Moreover, due to the light-tailed nature of log-concave distributions, w.h.p. every projected sample from any component F_i lies within a ball with small radius centered at the projected mean of F_i. Intuitively, these facts explain why all projected samples from a large component T_r are separated from those from other components. The authors prove that the point p_0 that maximizes $\sigma(p)$ must come from a large component F_r and that if it does, the output will be the set T_r exactly. This concludes the proof of correctness of the algorithm.

3 Heavy-Tailed Distributions

The paper [1] focuses on heavy-tailed distributions when some of the moments can even be infinite. In these cases, medians can serve as a more robust estimator than means or variances. Motivated by this, the median radius of a one-dimensional distribution is defined.

Definition 1. *Let X be a random variable with cumulative distribution function $F(x)$. The center of X is the minimum c such that $F(c) = \frac{1}{2}$. The radius R of X is the smallest R such that half of X's density is in the interval $[c - R, c + R]$. This definition is generalized to multidimensional distributions by considering the centers coordinate-wise.*

Let $\mathcal{F}_0$ be the class of distributions in $\mathbb{R}^n$ with independent coordinates, radius at most R and symmetric and monotonically decreasing tails. Let $\mathcal{F}_1$ be a subset of $\mathcal{F}_0$ with one additional constraint: for any distribution $D \in \mathcal{F}_1$ centered at μ and any coordinate D_i of D, a random sample x from D_i satisfies

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

where μ_i is the center of D_i. This last condition is rather tame, since it is satisfied by any distribution with finite variance as well as other families.

The authors show that for a mixture of k distributions $D_1, \cdots, D_k$ from $\mathcal{F}_0$ with centers at $\mu_1, \cdots, \mu_k$ such that $\|\mu_i - \mu_j\|_2 \geq \Omega\left(R\sqrt{\frac{k}{\epsilon}}\right)$ and $\frac{\|\mu_i - \mu_j\|_2}{\|\mu_i - \mu_j\|_\infty} \geq \Omega\left(\sqrt{\frac{k}{\epsilon}}\right)$ there is an algorithm that correctly classifies all but ϵ fraction of samples with high probability and uses a number of samples polynomial in $n, k, 1/\epsilon$ and $1/w_{\min}$. For mixtures from $\mathcal{F}_1$ a larger separation of the centers is required, but the slope condition is dropped. In particular, they require $\|\mu_i - \mu_j\|_2 \geq \Omega^*\left(\frac{Rk^{\frac{5}{2}}}{\epsilon^2}\right)$ in general and $\|\mu_i - \mu_j\|_2 \geq \Omega\left(\frac{Rk^2}{\epsilon^2}\right)$ for the easier case where centers are known.

The key insight of this algorithm is that the most elementary approach works: when centers are known, simply classify each point to the cluster whose center is nearest to it. The question in this case is how do we define "near". The surprising answer is that the ℓ_1 norm is sufficient. In fact, the authors show that there exist distributions for which the same algorithm and the ℓ_2 norm misclassify a constant fraction of the points. A key difference between these norms is that ℓ_2 is rotationally symmetric whereas ℓ_1 is not. This makes clear an implicit assumption of this model: the distributions we are considering are coordinate-wise independent. If they were spherically symmetric instead, both norms would produce the same results; the paper does not solve the problem in such case.

3.1 Main technical lemmas

The main technical lemma for learning mixtures with known centers comes from a robust property of arbitrary symmetric distributions. The condition basically says that a sample from the distribution is likely to be closer to the center of the distribution than the any fixed point sufficiently far from the center. The lemma further imposes some slope conditions on this fixed point, but the authors bypass this later for distributions from $\mathcal{F}_1$.

Lemma 2. *Let ϵ and C be constants and let D be a distribution in $\mathbb{R}^n$ centered at the origin with radius R. Let μ be a point such that $\|\mu\|_2 \geq 4R(C + \frac{1}{\sqrt{\epsilon}})$, and having a slope ratio $\frac{\|\mu\|_2}{\|\mu\|_\infty} \geq 4(C + \frac{1}{\sqrt{\epsilon}})$. A point x sampled from D will satisfy*

$$\|x - \mu\|_1 - \|x\|_1 \geq C\|\mu\|_2 \geq C^2 R$$

with probability at least $1-\epsilon$.

Lemma 3. *Fix* $\epsilon \leq \frac{1}{10}$. *Suppose* $D_1 \in \mathcal{F}_1$ *and* $\mu \in \mathbb{R}^n$ *satisfies* $\|\mu\|_2 \geq \frac{6000R}{\epsilon^2}$. *Then* x *sampled from* D_1 *will satisfy*

$$\|x-\mu\|_1 - \|x\|_1 \geq \frac{\|\mu\|_2}{15}$$

with probability at least $1-\epsilon$.

The proof of the Lemma 2 results from an application of Chebyshev's inequality with appropriate lower bounds on the expectation and upper bounds on the variance. The proof of Lemma 3 separates large and small coordinates by absolute values of μ_i with a threshold at $O(R/\epsilon)$. For the larger group, the additional restriction of $\mathcal{F}_1$ is used while the previous lemma is still valid for the smaller groups. Both results put together with union bound provide the following theorem.

Theorem 2. *Consider a mixture of* k *distributions* $D_1, \cdots, D_k$ *with known centers* $\mu_1, \cdots, \mu_k$. *If either of the following conditions is met, then classification according to nearest center in the* ℓ_1 *norm succeeds with probability at least* $1-\epsilon$.

- *For every* $i \neq j$, $\|\mu_i - \mu_j\|_2 \geq \Omega\left(R\sqrt{\frac{k}{\epsilon}}\right)$ *and* $\frac{\|\mu_i-\mu_j\|_2}{\|\mu_i-\mu_j\|_\infty} \geq \Omega\left(\sqrt{\frac{k}{\epsilon}}\right)$ *or*
- *Each distribution belongs to the class* $\mathcal{F}_1$ *and for every* $i \neq j$, $\|\mu_i - \mu_j\|_2 \geq \Omega\left(R\frac{k^2}{\epsilon^2}\right)$.

3.2 Algorithm for the general case

The intuition for the general case where the centers are not known is the following. If we knew the centers, then we could know that the ℓ_1 norm would be sufficient to classify the points. Consider partitioning the coordinates into two groups and clustering the points independently using both partitions, then we should get approximately the same clusters.

The algorithm iteratively selects a set of sample points S_0 to build the clusters and another group S_1 to cross-validate the clusters. All possible clusterings of S_0 into $k+1$ groups are considered. The algorithm then computes the median of each cluster except the last one for both partitions and then uses these centers to classify the points of S_1. If the first and second clusters are similar for each cluster and the $(k+1)$-st cluster is small, the algorithm accepts. Otherwise, it will pick another partition of the coordinates and try again.

At first, it might seem discouraging that the algorithm might have to go through every possible partition of the coordinates, but the authors quickly show that most partitions will be good provided that the centers are sufficiently not axis-aligned. The running time still is exponential in k though.

References

[1] Anirban Dasgupta, John Hopcroft, Jon Kleinberg, and Mark Sandler. On learning mixtures of heavy-tailed distributions. In *the 46th Annual IEEE Symposium on Foundations of Computer Science*, pages 491–500, Oct 2005.

[2] Ravindran Kannan, Hadi Salmasian, and Santosh Vempala. The spectral method for general mixture models. *SIAM Journal on Computing*, 38(3):1141–1156, 2008.

[3] Santosh Vempala and Grant Wang. A spectral algorithm for learning mixtures of distributions. In *Journal of Computer and System Sciences*, pages 113–123, 2002.