

Robust PCA and Clustering in Noisy Mixtures *

S. Charles Brubaker †

Abstract

This paper presents a polynomial algorithm for learning mixtures of logconcave distributions in \mathbb{R}^n in the presence of malicious noise. That is, each sample is corrupted with some small probability, being replaced by a point about which we can make no assumptions. A key element of the algorithm is Robust Principle Components Analysis (PCA), which is less susceptible to corruption by noisy points. While noise may cause standard PCA to collapse well-separated mixture components so that they are indistinguishable, Robust PCA preserves the distance between some of the components, making a partition possible. It then recurses on each half of the mixture until every component is isolated. The success of this algorithm requires only a $O^*(\log n)$ factor increase in the required separation between components of the mixture compared to the noiseless case.

1 Introduction

Clustering points in \mathbb{R}^n is common practice in a large variety of applied fields such as computer vision, robotics, speech recognition, web search, and spam filtering. Heuristics such as “k-means clustering” [17, 11] and “expectation maximization” [7] have been used in these fields since such computation became practical. Over the past few years, however, algorithms with theoretical guarantees have been discovered, where it is assumed that the points come from a mixture distribution, i.e. a convex combination of distributions of a known type. Mixtures of Gaussians or logconcave distributions and mixtures of product distributions have received the most attention. The goal of clustering is to group points generated by the same component together and separate points generated by different components.

This paper explores the case where the data includes some small miscellaneous component in addition to a well-behaved mixture. Equivalently, we may say that the sampling process for the well-behaved mixture has some noise, whereby with some small probability a point is replaced by a noise point about which can make no assumptions. The practical importance of robustness to the presence of noise should be apparent to

anyone who has tried to file his bills, organize a closet, or set up a directory structure on his hard disk. Some things just don't belong to any large category. The presence of these “noisy” objects does not usually impede our ability to cluster or classify objects, suggesting that we should hold our algorithms to this standard as well. More concretely, in tasks such as document or web-page clustering it is unreasonable to assume that components will be well-separated with *absolutely nothing* in-between.

Previous work has addressed noiseless version of the clustering problem for mixtures of logconcave distributions [13, 1]. Recall that a mixture density is a convex combination of distributions of a known type. It has the functional form

$$F(x) = w_1 f_1(x) + \dots + w_k f_k(x),$$

where f_i is from a special class of distributions, e.g. Gaussians. The coefficients w_i are called the mixing weights and sum to one. Sampling from a mixture distribution can be thought of as a two step process. First, a component index i is chosen according to the mixing weights. Second, a point x is sampled according to the distribution f_i . Given a collection of points sampled in this fashion, clustering is the task of partitioning the points according to the component that generated them. The set of points drawn from a particular component is called a cluster. The case where the dimension of the points n is much larger than the number of components k is the most interesting for both theoreticians and practitioners.

What distinguishes this paper from previous work is that it assumes that some of the sample points may have been corrupted in an arbitrary way. That is, with probability at most ϵ the sample source outputs a point about which we can make no assumptions. We call such a sample source ϵ -noisy. This is the natural analog to the malicious error models of [14, 20] for the clustering problem. Because the noise component is arbitrary, it may not be possible to cluster in the traditional sense. Indeed all noise points could be identical to one of the non-noise points, making them indistinguishable. Therefore, we set a different goal. Suppose the data set can be written as the disjoint union $S_1 \cup \dots \cup S_k \cup N$, where S_i corresponds to the set of points

*Supported in part by NSF award CCF-07 and a Raytheon fellowship

†Georgia Institute of Technology

from component i and N to the set of noise points. Then we seek a collection of disjoint sets $C_1 \dots C_k$ such that for every S_i , there is a unique C_i where

$$(1.1) \quad S_i \subseteq C_i \subseteq S_i \cup N.$$

Although the sets $\{C_i\}$ may include some noise points, they induce a correct partition of the non-noise points $\{S_i\}$.

A common approach for classifying points from mixture models is to project to the top k principal components [13, 1, 21]. This fails in the presence of noise. In fact, only k well-chosen malicious noise points are required to cause the intermean distances to become arbitrarily small after such a projection. For instance let v_{n-k+1}, \dots, v_n be the k smallest spectral components for a set of samples S . To this set add k noise points, $x_1 = cv_n, \dots, x_k = cv_{n-k+1}$. For large values of c , the largest k spectral components of $S \cup \{x_i\}_{i=1}^k$ will converge to $v_{n-k+1} \dots v_n$.

Methods for clustering points after projection fare little better amidst malicious noise. For instance, the single link method (used for example in Achlioptas and McSherry [1]) fails, as a large links in the graph can be subdivided through the insertion of noise points.

1.1 Results We present a polynomial time algorithm that given a noisy mixture of well-separated, logconcave distributions in \mathbb{R}^n , learns to separate the components of the mixture. That is, the algorithm finds a partition of \mathbb{R}^n into k sets with disjoint interiors each of which contains almost all of the probability mass of a unique component of the mixture. The error of such a partition is the total mass that falls outside of the correct set. As a corollary, this algorithm makes it possible to cluster points from a noisy source in the sense of Eqn. 1.1. The separation between the means necessary for the algorithm's success is only an $O^*(\log n)$ factor larger than the best analogous results without noise, treating k and w_{\min} as constants.

The input to our algorithm is a source of samples LM, a natural number k and a scalar w_{\min} . The quantity k is the number of non-noise components in the mixture and w_{\min} is a lower bound on the minimum mixing weight. For simplicity of the exposition, we state the results in terms of learning a partition of \mathbb{R}^n (i.e. a classifier). This can easily be turned into a statement strictly about clustering a set of points through a slight modification of the algorithm.¹

¹For instance, we might divide the given points into overlapping blocks and cluster each block using the remaining points to simulate the sample source. The overlap of the blocks can be used to calculate to appropriate permutation of component indices for each block and thus obtain a clustering of the whole set.

For a component i of the mixture, we define the following quantities : $\mu_i = E[x]$, $R_i^2 = E[\|x - \mu_i\|^2]$, and $\sigma_i^2 = \max_{v \in \mathbb{R}^n} E[(v \cdot (x - \mu_i))^2] / \|v\|^2$. We define the mixing weights w_i to be the probability that LM outputs a sample from component i . Thus, if LM outputs noise with probability ξ , then $\sum_{i=1}^k w_i = 1 - \xi$, effectively treating ξ as a mixing weight itself. We let w_{\min} be the minimum mixing weight.

Our main result is summarized in the following theorem.

THEOREM 1.1. *Let \mathcal{F} be a mixture of k logconcave distributions with means $\{\mu_i\}$ and maximum variances $\{\sigma_i\}$. Let $\delta, \eta > 0$. There exist $\epsilon = \Omega(w_{\min} \log^{-2}(nk/(w_{\min}\delta\eta)))$ and $\alpha = O(k^{3/2}w_{\min}^{-1} \log(nk/(w_{\min}\delta\eta)))$ such that if LM is an ϵ -noisy sample source for \mathcal{F} and if for every pair of components i, j*

$$(1.2) \quad \|\mu_i - \mu_j\| \geq \alpha(\sigma_i + \sigma_j),$$

then the following holds. There is a polynomial algorithm that given access to at least $O(nkw_{\min}^{-1} \log^6(nk/\delta))$ samples from LM with probability $1 - \delta$ returns a partition of \mathbb{R}^n that correctly classifies a point from \mathcal{F} with probability $1 - \eta$.

1.2 Previous Work The problem of learning mixture models was popularized by S. Dasgupta in [6], beginning with mixtures of spherical Gaussians. This result was later improved upon by S. Dasgupta and Shulman [19] and by Arora and Kannan [18] to require a separation of

$$\|\mu_i - \mu_j\| \geq O(n^{1/4})(\sigma_i + \sigma_j).$$

The latter work also handles a wider variety of cases, including the case where one Gaussian is "inside" another because of a large difference in their variances.

Vempala and Wang [21] showed that spectral projection preserves the intermean distances while dramatically reducing the intra-cluster distances. Their algorithm requires only

$$\|\mu_i - \mu_j\| \geq O^*(k^{1/4})(\sigma_i + \sigma_j)$$

separation for k components. This work was later extended in by Kannan, Salmasian, and Vempala [13] to mixtures of logconcave densities that are not necessarily isotropic. Achlioptas and McSherry [1] show similar results and explore the minimum necessary separation for clustering to be possible. In these works, the required separation is

$$\|\mu_i - \mu_j\| \geq O^*(k^{3/2} + w_{\min}^{-1/2})(\sigma_i + \sigma_j).$$

All of the above separation bounds are given in terms of σ_i^2 , the maximum directional variance. This is far stronger than what is required for separability, as the components may be separated along a direction where the variance are much smaller than σ_i^2 . Brubaker and Vempala [2] address this issue via an affine-invariant clustering algorithm. For two Gaussian components, the separation condition is nearly optimal for hyperplane separability, requiring only that there be *some* direction along which the distance between the means is large compared to the variance in this direction. For more than two components, the separation condition is stated in terms of the Fisher discriminant.

A related area of work is on learning product distributions, where the coordinates are independent (e.g. a Gaussian would be axis-aligned). Here the goal is not necessarily to cluster data but to approximate the density of the mixture. Freund and Mansour [10] first solved this problem for a mixture of two distributions of binary vectors, finding a model that approximates the true distribution in terms of Kullback-Leibler distance. Feldman and O'Donnell [8] extended this result to mixtures of any constant number of components and to discrete domains instead of binary vectors, i.e. $\{0, \dots, b-1\}^n$ instead of $\{0, 1\}^n$. Joined by Servedio in [9], they applied their technique to mixtures of a constant number of axis-aligned Gaussians, showing that they can be approximated without any separation assumption at all.

Another class of results on learning product distributions uses separation conditions which assume that the component centers be separated along many directions. Chaudhuri and Rao [4] note that results such as [13],[1] and [2] have a polynomial dependence on the inverse of the minimum mixing weight and reduce this to a logarithmic dependence by exploiting the independence of the coordinates. Beyond logconcave distributions, A. Dasgupta et al [5] consider a class of heavy-tailed product distributions and give separation conditions under which such distributions can be learned using an algorithm that is exponential in the number of samples. Chaudhuri and Rao [3] have recently given a polynomial algorithm to learn a related class of heavy-tailed product distributions.

The term ‘‘Robust PCA’’ has been used in previous work to describe several proposed algorithms in the fields of robust statistics, computer vision [15, 22] and bioinformatics [12]. Like our algorithm, these works attempt to find the PCA subspace in the presence of outliers. None, however, use the idea of alternating removing outliers and projecting to a lower dimensional subspace, as our algorithm does.

2 Algorithm

In previous work [13], the approach is to first project the data onto its top k spectral components, then extract a single cluster, and repeat. This strategy succeeds because the projection onto the top k components preserves much of the intermean distances, while reducing the pairwise distance between points of the same component. The concentration of the pairwise distances is then exploited to remove a component.

In the presence of noise, however, this approach breaks down. In fact, only k well-chosen noise points are required to cause the intermean distances to become arbitrarily small after projection. To cope with this problem we first remove outliers. That is, we reduce the maximum distance between any two points to be $O(R_{\max} + \mu_{\max})$. Projection to the top k components still may not preserve the necessary intermean distances, but projection to the top $\lfloor (n - k)/2 \rfloor + k$ components will. By repeating this procedure of first removing outliers and then projecting, we reduce the dimension to k . We call this procedure Robust PCA.

Robust PCA will preserve enough of the distance between the components whose means are furthest apart so that the direction between their means can be approximated by a pair of samples, one coming from each component. Imagine projecting the entire mixture density onto this line. The concentration of the individual components implies that this density will be multimodal with large peaks and long flat valleys. By setting a threshold in the middle of a valley, we define a hyperplane that separates the components of the mixture.

We then recurse on the two half-spaces defined by this hyperplane. At lower levels of the recursion tree, we are recursing on the intersection of these hyperplanes, i.e. a polyhedron. Ideally, we would like to recurse on a submixture, i.e. a subset of the original mixture’s components. Fortunately, each component is far enough away from the the support hyperplanes that the probability that a sample will appear on one side of a hyperplane while its component mean is on the other is vanishingly small. This enables us to simulate the desired submixture by rejecting samples until one from the correct part of \mathbb{R}^n is obtained.

2.1 Robust PCA We now describe our algorithm in more detail, considering Robust PCA first. We initialize the subspace W to be \mathbb{R}^n and then reduce the dimension of this subspace in each iteration until it has only k dimensions. Each iteration considers a fresh set of samples in step 2a to avoid dependency issues in the analysis. These points are projected to the current

subspace W . In steps 2b and 2c, we compute

$$(2.3) \quad t(Z) = s_r(\{s_r(\{\|p - q\| : q \in Z\}) : p \in Z\})$$

where s_r is the function which selects the r th largest element of a set. This computation is accomplished by arranging all pairwise distances into a matrix indexed by p and q . For every row, we select the r th largest element (step 2b), and from the resulting set we then select the r th largest element (step 2c). This gives a pair p_0, q_0 such that $\|p_0 - q_0\| = t(Z)$.

Next, we remove all points that are further than $\xi t = 16\beta t$ from p_0 to form the set Z' and compute the matrix

$$\sum_{p \in Z'} (p - p_0)(p - p_0)^T.$$

We then compute the SVD of this matrix and let the top $\ell = \lfloor (\dim(W) - k)/2 \rfloor + k$ singular vectors be the span of W . This process is repeated with fresh samples until $\dim(W) = k$.

Algorithm 1 Robust PCA

Input:

- 1) Collection $\{Z_i\}$ of $\lceil \log_2 n \rceil$ sets of points in \mathbb{R}^n .
- 2) Integers k, r , scalar ξ .

Output: A subspace W of dimension k .

1. Let $W = \mathbb{R}^n$.
 2. While $\dim(W) > k$,
 - (a) Let $Z = \text{proj}_W(Z_i)$, where Z_i is the next set of samples.
 - (b) For every $p \in Z$ find the point $q(p)$, defined to be r th furthest away point.
 - (c) Find the point p_0 such that the distance $\|p_0 - q(p_0)\|$ is the r th largest distance in the set $\{\|p - q(p)\| : p \in Z\}$. Let $q_0 = q(p_0)$ and let $t(Z) = \|p_0 - q_0\|$.
 - (d) Let $Z' = Z \cap B(p_0, \xi t(Z))$.
 - (e) Let W be the span of the top $\lfloor (\dim(W) - k)/2 \rfloor + k$ eigenvectors of the matrix $\sum_{p \in Z'} (p - p_0)(p - p_0)^T$.
-

2.2 Clustering Noisy Mixtures We now describe the algorithm for clustering noisy mixtures. For an input polyhedron P the goal is to separate the components whose means are contained in the polyhedron. We call this subset of components a submixture and to obtain samples from it we get points from the full mixture source LM and then intersect these points with the polyhedron P . This is done in steps 2 and 3. The analysis

will show that this effectively simulates sampling from the submixture.

The set of samples is used to call Robust PCA to find a subspace W of dimension k . Another set of points X is sampled in step 3 and used to compute the resolution d in step 4, which serves as the bucket diameter as we search for a partition which does not cut the mixture components. We then obtain an independent sample Y that includes at least one point from every component with high probability. For every pair of points (a, b) , we then attempt to separate the components along the direction $v = (a - b)/\|a - b\|$. To separate along a direction we project a set of points to v and place them in buckets of size d , which divide the real line. If there are two full buckets with at least $w_{\min} m_X/4$ and a nearly empty bucket with at most $2\epsilon m_X$ points between, then we call this set of buckets a valley. We place the dividing threshold γ in the middle of the nearly empty bucket in the valley. We then recurse on the two polyhedra $P \cap H_{v, \gamma}$ and $P \cap H_{-v, -\gamma}$, where $H_{v, \gamma}$ denotes the halfspace $\{x \in \mathbb{R}^n : v \cdot x \geq \gamma\}$.

If no valley is found in all of $Y \times Y$, then we conclude that a single component has already been isolated and simply return the current polyhedron. The algorithm finally returns all such polyhedra.

3 Preliminaries

In our analysis, we will decompose a set Z obtained from LM into $S \cup N$ where S consists of the points drawn from \mathcal{F} and N consists of the noise points. Further, we decompose the set S into $S_1 \cup \dots \cup S_k$, where S_i consists of the points drawn from component i . For a point $p \in S$, we use $\ell(p)$ to denote the component from which p was drawn. We also use $\hat{\mu}_i$ to indicate the average of points from component i in a set. For a subspace W and polyhedron P , it will be convenient to define the following quantities. Let $I_P = \{i : \mu_i \in P\}$ and let \mathcal{F}_P be the submixture consisting of the components in I_P . Let

$$\begin{aligned} R_i^{(W)} &= E_i [\|\text{proj}_W(x - \mu_i)\|^2]^{1/2} \\ R_{\max}^{(W, P)} &= \max_{i \in I_P} R_i^{(W)}. \\ \mu_{\max}^{(W, P)} &= \max_{i, j \in I_P} \|\text{proj}_W(\mu_i - \mu_j)\| \\ \sigma_{\max}^{(P)} &= \max_{i \in I_P} \sigma_i. \end{aligned}$$

Note that E_i denotes an expectation with respect to the i th component of the mixture. When the superscript W is omitted, it may be assumed that \mathbb{R}^n is meant. The polyhedron P is often clear from context and may be omitted as well.

Throughout the analysis we use the fact

Algorithm 2 Cluster Noisy logconcave Mixture

Input:

- 1) Sampling source LM which generates point in \mathbb{R}^n .
- 2) Integer k , reals ϵ, w_{\min} .
- 3) Polyhedron P . (Note $P = \mathbb{R}^n$ in the initial call.)

Output: A collection of k polyhedra.

1. For $i = 1$ to $\lceil \log n \rceil$ let Z_i a set m_Z points from LM.
 2. Let W be the subspace returned by Robust PCA for the collection $\{Z_i \cap P\}$, $\xi = 16\beta$ and $r = \lfloor 2\epsilon m_Z \rfloor$.
 3. Let $X = \text{proj}_W(X_0 \cap P)$, where X_0 is a set of m_X samples obtained from LM.
 4. Let $d = t(X)/10k$.
 5. Let $Y = \text{proj}_W(Y_0)$, where Y_0 is a set of m_Y samples from LM.
 6. For every $(a, b) \in Y \times Y$
 - (a) Let $v = (a - b)/\|a - b\|$.
 - (b) Let $b_i = |\{x \in X : \text{proj}_v(x) \in [id, (i + 1)d]\}|$.
 - (c) If there is a triple $i_1 < i_2 < i_3$ where $b_{i_1}, b_{i_3} > w_{\min} m_X/4$ and $b_{i_2} \leq 2\epsilon m_X$, then let $\gamma = (i_2 + 1/2)d$ and recurse with $P = P \cap H_{v, \gamma}$ and $P = P \cap H_{-v, -\gamma}$. Return the collection of polyhedra produced by these calls.
 7. Return P .
-

that the lower bound on the separation $\alpha = \Theta(k^{3/2} w_{\min}^{-1} \log(nk/(w_{\min} \delta \eta)))$ and the upper bound on the noise $\epsilon = \Theta(w_{\min} \log^{-2}(nk/(w_{\min} \delta \eta)))$.

3.1 Safe Polyhedra The success of the algorithm depends on the fact that intersecting the sample set from LM with the polyhedron P in steps 2 and 3, of Algorithm 2 effectively simulates sampling from the submixture \mathcal{F}_P . That is, this intersection has the effect of including all points from components in I_P and excluding all points from other components. This motivates the following definition.

DEFINITION 3.1. A polyhedron P is η -safe for a mixture \mathcal{F} if

1. For every $i \in I_P$, we have $\Pr[x \notin P] \leq \eta$, where x is a random point from component i .
2. For every $i \notin I_P$, we have $\Pr[x \in P] \leq \eta$, where x is a random point from component i .

The concentration of logconcave distributions yields a simple criterion for showing that a halfspace is safe. We use the following theorem from [16].

THEOREM 3.1. Let $R^2 = \max_{\|v\|=1} E[(v \cdot (x - \mu))^2]$ for a random variable x from a logconcave distribution. Then

$$\Pr(\|x - \mu\| > tR) < e^{-t+1}.$$

Restricting this to a single dimension gives the following corollary.

COROLLARY 3.1. Let $H_{v, \gamma} = \{x \in \mathbb{R}^n : v \cdot x \geq \gamma\}$ be a halfspace in \mathbb{R}^n . For every $\eta > 0$, there is a factor $\beta_{\text{safe}} = O(\log 1/\eta)$ such that if for every component i in a logconcave mixture \mathcal{F} ,

$$|v \cdot \mu_i - \gamma| > \beta_{\text{safe}} \sigma_i,$$

then $H_{v, \gamma}$ is η -safe for \mathcal{F} .

Halfspaces are then easily combined into polyhedra.

PROPOSITION 3.1. If P_1 is η_1 -safe for \mathcal{F} and P_2 is η_2 -safe for \mathcal{F}_{P_1} , then $P_1 \cap P_2$ is $(\eta_1 + \eta_2)$ -safe for \mathcal{F} .

3.2 Properties of Sample Sets As we will argue, the polyhedra obtained by the algorithm will be safe. Therefore, we expect that the polyhedra will contain the points from the components whose means are contained in the polyhedra. We also expect that no set chosen in steps 1 or 3 of Algorithm 2 will contain much more than its share of noise points and that the empirical means and variances will be close to those of the component distributions themselves. Our analysis rests on these sets obtained from LM in steps 1 and 3 having these and other key properties that are summarized in the following definition.

DEFINITION 3.2. A set $S_1 \cup \dots \cup S_n \cup N$ of m points from LM is good for subspace W , polyhedron P , and scalar β if the following conditions hold.

1. For every component i , if $\mu_i \in P$, then $S_i \subseteq P$, and if $\mu_i \notin P$, then $S_i \cap P = \emptyset$.
2. $|S_i| \geq w_i m/2$ for all components i , and $|N| \leq 2\epsilon m$.
3. For every component $i \in I_P$ and every $p \in S_i$, $\|\text{proj}_W(p - \mu_i)\| \leq \beta R_i^{(W)}$.
4. For every component $i \in I_P$ $\|\text{proj}_W(\mu_i - \hat{\mu}_i)\| \leq \frac{\sigma_i}{4}$.
5. For every component $i \in I_P$

$$\frac{7}{8} R_i^{(W)} \leq \frac{1}{|S_i|} \sum_{p \in S_i} \|\text{proj}_W(p - \hat{\mu}_i)\|^2 \leq \frac{8}{7} R_i^{(W)}.$$

6. For some pair $i, j \in I_P$ such that $\|\text{proj}_W(\mu_i - \mu_j)\| = \mu_{\max}^{(W, P)}$, it holds for all $p \in S_i \cup S_j$, that $\|\text{proj}_u(p - \mu_{\ell(p)})\| \leq \beta \sigma_{\max}^{(P)}$, where u is the unit vector along the direction $\text{proj}_W(\mu_i - \mu_j)$.

For convenience, we will sometimes say the set $Z = \text{proj}_W(Z_0 \cap P)$ is “generated by a good set for W, P , and β .” This is not really a property of the set Z itself, but rather of W, P , β and an implicit Z_0 (drawn from LM) such that $Z = \text{proj}_W(Z_0 \cap P)$.

It is important to note that a set is only good for a particular subspace. In our analysis on Robust PCA, we will require that Z_i from step 1 of Algorithm 2 be good for polyhedron P and the current subspace W in step 2a of Robust PCA, where $Z_i \cap P$ is used. Thus, Z_1 must be good for \mathbb{R}^n and Z_2 must be good for the subspace obtained after one iteration in Robust PCA, etc. Finally, the set X_0 used in step 3 of Algorithm 2 must be good for W_k the subspace returned by Robust PCA. The following lemma shows that this happens with high probability.

LEMMA 3.1. *Let Z_0 be a set of m points generated by ϵ -noisy sample source LM for a logconcave mixture. Let P be polyhedron that is $(\delta/2m)$ -safe. Let W be any subspace of \mathbb{R}^n . There exist $M_{\text{good}} = O(n/w_{\min} \log^5 nk/\delta)$ and $\beta_{\text{good}} = O(\log(mk/\delta))$ such that with probability $1 - \delta$ if $m \geq M_{\text{good}}$, then Z_0 is good for W , P , and any $\beta \geq \beta_{\text{good}}$.*

Proof. We consider the goodness properties in order. From the definition of η -safe, item 1 holds with probability $1 - \delta/2$. Item 2 follows from a Chernoff bound (recall that ϵ is an upper bound on the noise of LM and not the noise itself). The remaining items are standard results for logconcave distributions. See [13].

3.3 Bounds on t It is important that we be able to approximate the greatest distance between two non-noise points. This enables us to put a ball around the non-noise data in Robust PCA so as to remove noise points that are far away from the non-noise points (step 2d of Robust PCA). It is also critical in determining d the resolution at which we look for valleys in the partitioning phase (step 4 of the clustering algorithm).

LEMMA 3.2. *Suppose that $Z = S \cup N$ was generated by a good set from LM for W , P , and β . Then $t = t(Z)$ has the bounds*

$$\begin{aligned} \max\{\mu_{\max}^{(W,P)} - 2\beta\sigma_{\max}^{(P)}, R_{\max}^{(W,P)}/2\} &\leq t \\ &\leq \max_{p,q \in S} \|p - q\| \leq \mu_{\max}^{(W,P)} + 2\beta R_{\max}^{(W,P)}. \end{aligned}$$

3.4 A Spectral Lemma For a matrix A , let $\lambda_j(A)$ be the j th largest eigenvalue of the matrix. When the matrix is clear from context, we may simply write λ_j . The following lemma will be useful in our analysis of Robust PCA.

LEMMA 3.3. *Let $A = M + C$ where M and C are symmetric positive semi-definite $n \times n$ matrices and $\text{rank}(M) = k$. Then for $j > k$,*

$$\lambda_j(A) \leq \frac{1}{j-k} \sum_{i=1}^j \lambda_i(C).$$

4 Analysis

We now turn to the major portion of our analysis. In Section 4.1, we analyze the effect of Robust PCA, showing that it preserves much of the distance between at least two means. In Section 4.3, we show the correctness of the partitioning step. Finally, we synthesize the whole argument in Section 4.2 to give the main theorem.

The essential parameters of the algorithm and analysis are $m_Z, m_X, m_Y, \beta, \epsilon$, and α . In terms of the quantities $n, w_{\min}, \delta, \eta$, and k , these are

$$\begin{aligned} m_Z &= C_Z n w_{\min}^{-1} \log^5(nk/\delta) \\ m_X &= C_X n w_{\min}^{-1} \log^5(nk/\delta) \\ m_Y &= C_Y w_{\min}^{-1} \log(k/\delta) \\ \beta &= C_\beta \log((m_X + m_Y + m_Z)k \log(n)/(\delta\eta)) \\ &= O(\log(nk/(w_{\min}\delta\eta))) \\ \epsilon &= C_\epsilon w_{\min}/\beta^2 \\ &= \Omega(w_{\min} \log^{-2}(nk/(w_{\min}\delta\eta))) \\ \alpha &= C_\alpha k^{3/2} w_{\min}^{-1} \beta \log n \\ &= O(k^{3/2} w_{\min}^{-1} \log(nk/(w_{\min}\delta\eta))) \end{aligned}$$

where the leading factor is an appropriate constant. We will exercise the choice of these constants in the course of the analysis. The reader will find it useful to refer to these equations in following the proof. Without loss of generality, we may assume that η is a polynomial factor smaller than δ .

4.1 Robust PCA Here we show that Robust PCA preserves most of the distance between the two components that are furthest part (Lemma 4.1). We accomplish this by showing that only a small fraction of this distance is lost as the dimension of the data is halved in each iteration (Lemma 4.2).

This result rests on two key claims. Claim 4.1 shows that the diameter of the non-noise data can be approximated in the presence of noise and that this permits the algorithm to place a relatively tight ball around the non-noise data, excluding noise points that are far away. The estimated diameter (roughly the parameter t) can neither be too small (or non-noise points will be excluded), nor too large (or noise points at the edge of the ball may have too large of an effect on the eigenvectors).

The other key claim is Claim 4.2 which bounds the maximum variance of the data in the subspace that is thrown out in an iteration. Recall that Robust PCA projects to $\lfloor (\dim(W) - k)/2 \rfloor + k$ dimensions, removes outliers outside the ball $B(p_0, 16\beta t)$, and repeats until a k dimensional subspace is found.

To illustrate why simply removing outliers and using standard PCA to project to a k dimensional subspace is inadequate, we define the following matrices. Let N' be the set of noise points after outliers are removed and assume that no non-noise points are removed. The remaining points are therefore $S \cup N'$. Assume p_0 be the origin, that $W = \mathbb{R}^n$ and consider the matrix computed in step 2e of Robust PCA

$$A = \frac{1}{m'} \sum_{p \in S \cup N'} pp^T,$$

where N' consists of the noise points that were not removed and $m' = |S \cup N'|$. Using the sample means $\hat{\mu}_i$ and covariance $\hat{\Sigma}_i$, we can decompose this matrix as the sum of

$$\begin{aligned} M &= \frac{1}{m'} \sum_{i=1}^k |S_i| \hat{\mu}_i \hat{\mu}_i^T \\ C &= \frac{1}{m'} \sum_{i=1}^k |S_i| \hat{\Sigma}_i \\ E &= \frac{1}{m'} \sum_{p \in N'} pp^T. \end{aligned}$$

The matrix M is the mixture of the outer product of the means, C is the mixture of the covariances, and E is the noise contribution.

Without noise, the second moment matrix A is just $M + C$. The rank of M is k and its eigenvectors are the subspace that we would ideally like to find, i.e. the span of the means. The matrix C can be viewed as a perturbation, which may cause the eigenvectors of $M + C$ to differ from those of M . The 2-norm of C is bounded from above by σ_{\max}^2 , while the 2-norm of M is bounded from below in terms of μ_{\max}^2 . For an adequate separation of the component means the matrix M dominates so that applying PCA to $M + C$ gives a k dimension subspace that is close to the span of the means.

In the presence of noise, however, we must account not only for the perturbation caused by C but that caused by E (the noise component) as well. At first, it may seem that the noise component cannot have a large effect. As we will show in the proof of Claim 4.2, the sum of the eigenvalues of E is comparable to that of C . Recall that the 2-norm is the largest

eigenvalue. While the sum of the eigenvalues of C may be on the order of $n\sigma_{\max}^2$, this is spread out over all n eigenvectors, so that no one eigenvalue is larger than σ_{\max}^2 . We have no such guarantee for E ; the sum may be concentrated in a single eigenvalue and therefore a single eigenvector. Even worse, it could be spread out over a constant fraction of the eigenvectors, each challenging the dominance of the eigenvectors of M . Hence, some constant fraction of the dimension must be preserved in order to avoid removing the eigenvectors of M , i.e. the span of the means. Claim 4.2 shows that half of the dimension is adequate to preserve most of the distance between the means.

In our analysis, we often will identify a subspace by giving its dimension as a subscript. For instance, we will use W_k to denote the subspace returned by Robust PCA and W_ℓ for intermediate subspaces within Robust PCA. The main result of this section is the following lemma.

LEMMA 4.1. *Suppose that every set Z obtained in step 2a of Robust PCA was generated by a good set for the current subspace W , P , and β . Then, letting W_k be the final subspace,*

$$\mu_{\max}^{(W_k, P)^2} \geq \frac{1}{2} \mu_{\max}^{(P)^2}.$$

Proof. This lemma is proved by applying the following lemma to each successive projection, until $n = k$.

LEMMA 4.2. *Let P be a polyhedron in \mathbb{R}^n and let W be a subspace in \mathbb{R}^n with dimension greater than k , where $\mu_{\max}^{(W, P)} \geq \mu_{\max}^{(P)}/2$. Suppose that the set Z in step 2a of Robust PCA is generated by a good set for W, P and β . Let W_ℓ be the subspace of dimension $\lfloor (\dim(W) - k)/2 \rfloor + k$ obtained in step 2e. Then for all pairs of components $i, j \in I_P$*

$$\begin{aligned} \|\text{proj}_{W_\ell}(\mu_i - \mu_j)\|^2 &\geq \|\text{proj}_W(\mu_i - \mu_j)\|^2 \\ &- (\lfloor (\dim(W) - k)/2 \rfloor + 1)^{-1} \frac{1}{4} \mu_{\max}^{(P)^2} - \frac{32k}{w_{\min}} \sigma_{\max}^{(P)^2}. \end{aligned}$$

Let μ_i and μ_j be the means of two components such that $\|\mu_i - \mu_j\| = \mu_{\max}^{(P)}$. We observe that the quantity $\mu_{\max}^{(W, P)}$ can only decrease as the dimension of W is reduced. Therefore, when we unravel the recurrence relation implied by Lemma 4.2, we may simplify the bound to

$$\begin{aligned} \|\text{proj}_{W_k}(\mu_i - \mu_j)\|^2 &\geq \|\mu_i - \mu_j\|^2 - \\ &\frac{\mu_{\max}^{(P)^2}}{8} \sum_{j=0}^{\lceil \log_2(n-k) \rceil} \frac{1}{2^j} - \sum_{j=0}^{\lceil \log_2(n-k) \rceil} \frac{32}{w_{\min}} k \sigma_{\max}^{(P)^2}, \end{aligned}$$

By our choice of the pair μ_i, μ_j , we have $\|\mu_i - \mu_j\|^2 = \mu_{\max}^{(P)2}$. Clearly, the first sum is bounded by $\mu_{\max}^{(P)2}/4$. The second sum becomes $32kw_{\min}^{-1}\sigma_{\max}^2\lceil\log_2(n-k)\rceil$. By the choice of α in Theorem 1, however, we may assume that this is no larger than $\mu_{\max}^{(P)2}/4$ either. Thus,

$$\mu_{\max}^{(W,P)2} \geq \|\text{proj}_{W_k}(\mu_i - \mu_j)\|^2 \geq \frac{1}{2}\mu_{\max}^{(P)2}.$$

Proof. [Proof of Lemma 4.2] Let $Z = S \cup N$ be a set generated by a good set of m_Z samples from LM. Let p_0, q_0 be a pair of points in $S \cup N$ satisfying $\|p_0 - q_0\| = t$. Because the denoising step removes all points outside of the ball $B(p_0, 16\beta t)$, we define the sets of remaining points $S' = S \cap B(p_0, 16\beta t)$ and $N' = N \cap B(p_0, 16\beta t)$. For convenience, we define $m' = |S' \cup N'|$.

We first claim that no non-noise points are eliminated (i.e. $S = S'$) and give a bound on the radius of the ball $B(p_0, 16\beta t)$.

CLAIM 4.1. *Suppose $p_0, q_0 \in S \cup N$ satisfy $\|p_0 - q_0\| = t$. Then*

$$S \subseteq B(p_0, 16\beta t) \subseteq B(p_0, 32\beta^2(\mu_{\max} + R_{\max})).$$

Thus, the second moment matrix used for the spectral analysis becomes

$$A = \frac{1}{m'} \sum_{p \in S \cup N'} (p - p_0)(p - p_0)^T.$$

This matrix has the following critical property.

CLAIM 4.2. *Suppose that $p_0, q_0 \in S \cup N$ satisfy $\|p_0 - q_0\| = t$. Then for $\ell = \lfloor (\dim(W) - k)/2 \rfloor + k$,*

$$\lambda_{\ell+1}(A) \leq \frac{w_{\min}}{64} (\lfloor (\dim(W) - k)/2 \rfloor + 1)^{-1} \mu_{\max}^{(W,P)2} + 4k\sigma_{\max}^{(P)2}.$$

By definition W_ℓ is the span of the top ℓ components of the matrix A . Let \bar{W}_ℓ be the complementary subspace in W . Consider a pair of means μ_i, μ_j . We will establish an upper bound on $\|\text{proj}_{\bar{W}_\ell}(\mu_i - \mu_j)\|^2$ and thus a lower bound on

$$(4.4) \quad \|\text{proj}_{W_\ell}(\mu_i - \mu_j)\|^2 = \|\text{proj}_W(\mu_i - \mu_j)\|^2 - \|\text{proj}_{\bar{W}_\ell}(\mu_i - \mu_j)\|^2$$

to prove the lemma.

Let v denote unit vector in the direction of $\text{proj}_{\bar{W}_\ell}(\mu_i - \mu_j)$. Let $e = \mu_i - \mu_j - (\hat{\mu}_i - \hat{\mu}_j)$. Then

$$\begin{aligned} \|\text{proj}_{\bar{W}_\ell}(\mu_i - \mu_j)\|^2 &= (v^T(\mu_i - \mu_j))^2 \\ &\leq 2(v^T(\hat{\mu}_i - \hat{\mu}_j))^2 + 2\|e\|^2. \end{aligned}$$

By item 4 of Definition 3.2, the term $\|e\|^2$ is bounded from above by $\sigma_{\max}^{(P)2}/4$. To bound the remaining term we use the fact that

$$1 \leq \frac{2}{w_{\min}} \frac{|S_i|}{m} \leq \frac{2}{w_{\min}} \frac{|S_i|}{m'}$$

from item 2 of Definition 3.2 to argue

$$\begin{aligned} (v^T(\hat{\mu}_i - \hat{\mu}_j))^2 &\leq 2((v^T(\hat{\mu}_i - p_0))^2 + (v^T(\hat{\mu}_j - p_0))^2) \\ &\leq 2 \sum_{i=1}^k (v^T(\hat{\mu}_i - p_0))^2 \\ &\leq \frac{4}{w_{\min}} \cdot \frac{1}{m'} \sum_{i=1}^k |S_i| (v^T(\hat{\mu}_i - p_0))^2. \end{aligned}$$

For each i ,

$$|S_i| (v^T(\hat{\mu}_i - p_0))^2 \leq \sum_{p \in S_i} (v^T(p - p_0))^2.$$

Including the points from N' , we then have

$$\begin{aligned} \frac{1}{m'} \sum_{i=1}^k |S_i| (v^T(\hat{\mu}_i - p_0))^2 &\leq \\ v^T \left(\frac{1}{m'} \sum_{p \in S \cup N'} (p - p_0)(p - p_0)^T \right) v &\leq \lambda_{\ell+1}(A). \end{aligned}$$

since v is in the subspace \bar{W}_ℓ .

From Claim 4.2, we have

$$\begin{aligned} \|\text{proj}_{\bar{W}_\ell}(\mu_i - \mu_j)\|^2 &\leq \frac{4}{w_{\min}} \lambda_{\ell+1}(A) + \frac{\sigma_{\max}^{(P)2}}{4} \\ &\leq \lfloor (\dim(W) - k)/2 \rfloor^{-1} \frac{1}{8} \mu_{\max}^{(W,P)2} + \frac{32}{w_{\min}} k \sigma_{\max}^{(P)2}. \end{aligned}$$

Combined with Eqn. 4.4, this proves the lemma.

Proof. [Proof of Claim 4.1] Since Lemma 4.2 assumes that $\mu_{\max}^{(W,P)} \geq \mu_{\max}^{(P)}/2$, we have

$$2\beta\sigma_{\max}^{(P)} \leq \alpha\sigma_{\max}^{(P)}/4 \leq \mu_{\max}^{(P)}/4 \leq \mu_{\max}^{(W,P)}/2,$$

using a suitable α and the separation assumption of Eqn. 1.2.

By Lemma 3.2 then

$$t \geq \mu_{\max}^{(W,P)} - 2\beta\sigma_{\max}^{(P)} \geq \mu_{\max}^{(W,P)}/2.$$

Also, by the same lemma $t \geq R_{\max}^{(W,P)}/2$. Without loss of generality assume $\beta \geq 1$. Then

$$\max_{p, q \in S} \|p - q\| \leq \mu_{\max}^{(W,P)} + 2\beta R_{\max}^{(W,P)} \leq 2t + 4\beta t < 8\beta t.$$

Thus, no two points in S can be further than $8\beta t$ apart.

Now let p be an arbitrary point in S and let $q \in S \cap B(p_0, t)$. Note that such a point q exists because by definition of t , $B(p_0, t)$ contains $(1 - 2\epsilon m) > |N|$ points. We have by the triangle inequality and Lemma 3.2 that

$$\begin{aligned} \|p_0 - p\| &\leq \|p_0 - q\| + \|q - p\| \leq t + 8\beta t \\ &< 16\beta t \leq 32\beta^2(\mu_{\max}^{(W,P)} + R_{\max}^{(W,P)}). \end{aligned}$$

Thus, $S \subseteq B(p_0, 16\beta t) \subseteq B(p_0, 32\beta^2(\mu_{\max} + R_{\max}))$.

Proof. [Proof of Claim 4.2] Without loss of generality, let us assume that p_0 is the origin. Thus, the matrix from line 6 of the algorithm becomes $A = \frac{1}{m'} \sum_{p \in S \cup N'} pp^T$. Using the sample means $\hat{\mu}_i$, we can decompose this matrix as the sum of

$$\begin{aligned} M &= \frac{1}{m'} \sum_{i=1}^k |S_i| \text{proj}_W(\hat{\mu}_i) \text{proj}_W(\hat{\mu}_i)^T \\ C &= \frac{1}{m'} \sum_{i=1}^k \sum_{p \in S_i} \text{proj}_W(p - \hat{\mu}_i) \text{proj}_W(p - \hat{\mu}_i)^T \\ E &= \frac{1}{m'} \sum_{p \in N'} pp^T. \end{aligned}$$

Our strategy will be to bound $\sum_{i=1}^{\ell+1} \lambda_i(C + E)$ and apply Lemma 3.3 to bound $\lambda_{\ell+1}(A)$.

$$\begin{aligned} \sum_{i=1}^n \lambda_i(C) &\leq \frac{1}{m'} \sum_{i=1}^k \sum_{p \in S_i} \|\text{proj}_W(p - \hat{\mu}_i)\|^2 \\ &= \sum_{i=1}^k \frac{|S_i|}{m'} \frac{1}{|S_i|} \sum_{p \in S_i} \|\text{proj}_W(p - \hat{\mu}_i)\|^2 \\ &\leq \max_i \frac{1}{|S_i|} \sum_{p \in S_i} \|\text{proj}_W(p - \hat{\mu}_i)\|^2 \\ &\leq \frac{8}{7} R_{\max}^{(W,P)^2}. \end{aligned}$$

By Claim 4.1, $N' \subseteq B(0, 16\beta t) \subseteq B(32\beta^2(\mu_{\max}^{(W,P)} + R_{\max}^{(W,P)}))$, so

$$\begin{aligned} \sum_{i=1}^n \lambda_i(E) &\leq 2\epsilon \max_{p \in N'} \|p\|^2 \leq \\ \epsilon 64\beta^2(\mu_{\max}^{(W,P)^2} + R_{\max}^{(W,P)^2}) &\leq \frac{w_{\min}}{64}(\mu_{\max}^{(W,P)^2} + R_{\max}^{(W,P)^2}), \end{aligned}$$

for an appropriate choice of $\epsilon = C_\epsilon w_{\min} \beta^{-2}$ from

Theorem 1.1. Combining these bounds

$$\begin{aligned} \sum_{i=1}^n \lambda_i(C) + \lambda_i(E) &\leq \\ \frac{w_{\min}}{64} \mu_{\max}^{(W,P)^2} + \left(\frac{w_{\min}}{64} + \frac{8}{7}\right) R_{\max}^{(W,P)^2} &\leq \\ \frac{w_{\min}}{64} \mu_{\max}^{(W,P)^2} + 2 \dim(W) \sigma_{\max}^2. \end{aligned}$$

Lemma 3.3 then gives the bound

$$\begin{aligned} \lambda_{\ell+1}(A) &\leq \frac{1}{\ell + 1 - k} \sum_{i=1}^{\ell+1} \lambda_i(C + E) \\ &\leq \frac{1}{\ell + 1 - k} \sum_{i=1}^n \lambda_i(C) + \lambda_i(E) \\ &\leq (\lfloor (\dim(W) - k)/2 \rfloor + 1)^{-1} \\ (4.5) \quad &\left(\frac{w_{\min}}{64} \mu_{\max}^{(W,P)^2} + 2 \dim(W) \sigma_{\max}^2 \right). \end{aligned}$$

We note that $2 \dim(W) / (\lfloor (\dim(W) - k)/2 \rfloor + 1) \leq 4k$. If $\dim(W) < 2k$, then this is trivial. On the other hand, if $\dim(W) \geq 2k$, then

$$\frac{2 \dim(W)}{\lfloor (\dim(W) - k)/2 \rfloor + 1} \leq \frac{4 \dim(W)}{\dim(W) - k} \leq 8 \leq 4k.$$

The bound of Eqn. 4.5 then becomes

$$\begin{aligned} \lambda_{\ell+1}(A) &\leq (\lfloor (\dim(W) - k)/2 \rfloor + 1)^{-1} \frac{w_{\min}}{64} \mu_{\max}^{(W,P)^2} \\ &\quad + 4k \sigma_{\max}^2. \end{aligned}$$

4.2 Partitioning Components We show that Algorithm 2 successfully partitions the components. The algorithm tries many directions in the subspace W until it finds a one with a ‘‘valley’’ corresponding to the intuition given in Section 2. We capture this notion formally in the following definition.

DEFINITION 4.1. *Let X be a set of m_X points in \mathbb{R} . For $i \in \mathbb{Z}$ let $b_i = |\{x \in X : x \in (id, (i+1)d]\}|$. We say that X has a valley if there is a triple $i_1 < i_2 < i_3$ such that $b_{i_1}, b_{i_3} > w_{\min} m_X / 8$ and $b_{i_2} \leq 2\epsilon m_X$. We define the point $d(i_2 + 1/2)$ to be the middle of the valley.*

Assuming that d is well-chosen, the existence of a valley is ensured by the fact that the means are well-separated compared to the width of the widest component. If d is too small, then we are likely to find valleys within the point set of a single component. If d is too large, then the whole mixture might fit into a single unit of resolution or ‘‘bucket.’’ When d is chosen correctly, non-noise points from two components fill the outer buckets, while only noise points fill the middle one.

The following two Lemmas show that with high probability the algorithm succeeds in a given node in the recursion tree. The first applies for internal nodes in the tree where components need to be separated, the second applies to the leaves where the division of space terminates.

LEMMA 4.3. *Let $\delta > 0$. Suppose that P is η -safe for \mathcal{F} and that $|I_P| > 1$. Then with probability $1 - \delta$, the halfspaces $H_{v,\gamma}$ and $H_{-v,-\gamma}$ are (η/k) -safe and each contains at least one component mean of I_P .*

LEMMA 4.4. *Let $\delta > 0$. Suppose that P is η -safe for \mathcal{F} and that $|I_P| = 1$. Then with probability $1 - \delta$, no valley will be found by Algorithm 2 in step 6c.*

4.3 Proof of the Main Theorem

Proof. [Proof of Theorem 1.1] Consider one node in the recursion tree of Algorithm 2, and suppose that P has $j < k$ support hyperplanes and that P is $(\eta j/k)$ -safe for \mathcal{F} . Note that in the root of the tree this is true because \mathbb{R}^n is 0-safe. In the case where the polyhedron contains more than one component mean (i.e. $|I_P| > 1$), Lemma 4.3 shows that with probability $1 - \delta'$ the half-space $H_{v,\gamma}$ obtained in line 9 excludes at least one component mean and is (η/k) -safe for \mathcal{F}_P . Proposition 3.1 then shows that $P \cap H_{v,\gamma}$ is $(\eta(j+1)/k)$ -safe for \mathcal{F} .

On the other hand, if the polyhedron P contains only one mean (i.e. $|I_P| = 1$), then with probability $1 - \delta'$ the algorithm does not find a valley and returns the polyhedron by Lemma 4.4. Thus, with probability $1 - 2k\delta'$ the algorithm returns a set of k polyhedra, each containing exactly one component mean and each η -safe for \mathcal{F} . Thus, we have by definition of η -safe that the collection of polyhedra induce a classifier that is correct with probability $1 - \eta$, as the theorem claims.

Acknowledgments

This research began as a class project with Vladimir Urazov. The author is grateful to Santosh Vempala for his feedback.

References

- [1] D. Achlioptas and F. McSherry. On spectral learning of mixtures of distributions. In *Proc. of COLT*, 2005.
- [2] S. C. Brubaker and S. Vempala. Isotropic pca and affine-invariant clustering. In M. Grötschel and G. Katoana, editors, *Building Bridges Between Mathematics and Computer Science*, volume 19 of *Bolyai Society Mathematical Studies*, 2008.
- [3] K. Chaudhuri and S. Rao. Beyond gaussians: Spectral methods for learning mixtures of heavy-tailed distributions. In *Proc. of COLT*, 2008.
- [4] K. Chaudhuri and S. Rao. Learning mixtures of product distributions using correlations and independence. In *Proc. of COLT*, 2008.
- [5] A. Dasgupta, J. Hopcroft, J. Kleinberg, and M. Sandler. On learning mixtures of heavy-tailed distributions. In *Proc. of FOCS*, 2005.
- [6] S. DasGupta. Learning mixtures of gaussians. In *Proc. of FOCS*, 1999.
- [7] A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *JRSS B*, 39:1–38, 1977.
- [8] J. Feldman and R. O’Donnell. Learning mixtures of product distributions over discrete domains. *Siam J. or Computing*, 37(5):1536–1564, 2008.
- [9] J. Feldman, R. A. Servedio, and R. O’Donnell. Pac learning axis-aligned mixtures of gaussians with no separation assumption. In *COLT*, pages 20–34, 2006.
- [10] Y. Freund and Y. Mansour. Estimating a mixture of two product distributions. In *COLT*, pages 53–62, 1999.
- [11] J. A. Hartigan and M. A. Wong. A k-means clustering algorithm. *Applied Statistics*, 28(1):100–108, 1979.
- [12] M. Hubert and S. Engelen. Robust pca and classification in biosciences. *Bioinformatics*, 20(11):1728–1736, 2004.
- [13] R. Kannan, H. Salmasian, and S. Vempala. The spectral method for general mixture models. *Siam J. Computing*, 38(3):1141–1156, 2008.
- [14] M. Kearns and M. Li. Learning in the presence of malicious errors. *SIAM J. on Computing*, 22(4):807–837, 1993.
- [15] F. De la Torre and M.J. Black. A framework for robust subspace learning. *IJCV*, 54:117–142, 2004.
- [16] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Struct. Algorithms*, 30(3):307–358, 2007.
- [17] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297, 1967.
- [18] R. Kannan S. Arora. Learning mixtures of arbitrary gaussians. *Ann. Appl. Probab.*, 15(1A):69–92, 2005.
- [19] L. Schulman S. DasGupta. A two-round variant of em for gaussian mixtures. In *Sixteenth Conference on Uncertainty in Artificial Intelligence*, 2000.
- [20] L. G. Valiant. Learning disjunction of conjunctions. In *IJCAI*, pages 560–566, 1985.
- [21] S. Vempala and G. Wang. A spectral algorithm for learning mixtures of distributions. *Proc. of FOCS 2002; JCCS*, 68(4):841–860, 2004.
- [22] L. Xu and A.L. Yuille. Robust principal component analysis by self-organizing rules based on statistical physics approach. *Trans. on Neural Networks*, 6(1):131–143, 1995.