Robust Learning of Mixtures of Gaussians

Daniel M. Kane

1Departments of CS/Math
University of California, San Diego
dakane@ucsd.edu

January 12th, 2021
Robust Statistics

Question

What statistical questions can we still answer if a small fraction of our data is (adversarially) corrupted?

- First considered by [Huber '64].
- Classical understanding of the information-theoretic limits.
- Until recently, algorithms for high dimensional problems either:
  - Had error that scaled polynomially in the dimension.
  - Had runtime that scaled exponentially in the dimension.
Algorithmic Robust Statistics

There was a breakthrough in 2016 with two papers [Diakonikolas-Kamath-K-Li-Moitra-Stewart, Lai-Rao-Vempala] that gave first polynomial time algorithms with dimension independent error. Since then there has been substantial progress:

- Mean estimation with bounded covariance [DKKLMS17]
- General Gaussians [DKKLMS16]
- Mixtures of two Binary Products [DKKLMS16]
- Mixtures of bounded number of spherical Gaussians [DKKLMS16]
- Sparse mean estimation [Balakrishnan-Du-Li-Singh ’17]
- Learning threshold functions [Diakonikolas-K-Stewart ’18]
- Stochastic optimization [Diakonikolas-Kamath-K-Li-Steinhardt-Stewart ’19]
Techniques

- Learn parameters in appropriate metric.
- Detect if empirical estimates are off by looking at higher moments.
- Use discrepancy of higher moments to remove outliers.
Gaussian Mixtures

Question

Robustly learn mixtures of arbitrary Gaussians.

This was resolved only recently in the non-robust case by [Moitra-Valiant '10, and by Belkin-Sinha '10]. Technique involved computing many moments of the mixture to high accuracy.
Obstacles to Robust Learning

The mixtures of Gaussians problem has long been considered a challenge problem for robust statistics. The standard techniques face the following obstacles:

- What metric do we learn parameters in?
- How do we estimate higher moments?
- How do we robustly learn distribution from moments?
Our Results

Theorem

Let $X = (G_1 + G_2)/2$ with $G_i = \mathcal{N}(\mu_i, \Sigma_i)$ be an equally weighted mixture of two general Gaussians in $\mathbb{R}^d$. For $\epsilon > 0$ sufficiently small, there exists an algorithm that given $\text{poly}(d/\epsilon)$, $\epsilon$-corrupted samples from $X$ and in time $\text{poly}(d/\epsilon)$ learns $X$ in total variational distance to error $\text{poly}(\epsilon)$.

This is the first algorithm to efficiently learn mixtures of arbitrary Gaussians robustly.
Strategy

- Deal with separated case.
- Normalize.
- Estimate moments.
- Tensor decomposition.
- Robust tournament.
Lemma (Robust Tournament, version in [DKKLMS16])

Given explicit distributions $X_1, \ldots, X_n$ and $O(\log(n)/\epsilon^2)$ $\epsilon$-corrupted samples from an unknown distribution $X$, there is a $\text{poly}(n/\epsilon)$ time algorithm that finds an $m$ so that $d_{TV}(X, X_m) = O(\min_{1 \leq i \leq n} d_{TV}(X, X_i) + \epsilon)$.

Proposition

Given $\epsilon$-corrupted samples from $X$, there is an efficient algorithm that for some $C, c > 0$ with probability at least $\epsilon^C$ returns a distribution $\hat{X}$ so that $d_{TV}(X, \hat{X}) < \epsilon^c$.

Combining these gives our final theorem.

- Our algorithm only needs to work with small probability. It can split into cases, make guesses.
Clustering

First, we deal with the case where the components are separated, namely where $d_{TV}(G_1, G_2) > 1 - \epsilon^c$. This is already solved by recent work!

**Theorem ([Bakshi-Kothari '20, Diakonikolas-Hopkins-K-Karmalkar '20])**

*Given $\epsilon$-corrupted samples from an equally weighted mixture $X$ of $k$ Gaussians that are pairwise $(1 - \epsilon)$-separated in total variational distance, there is an algorithm that uses $\text{poly}_k(d/\epsilon)$ time and samples and learns $X$ to error $\text{poly}_k(\epsilon)$.***

So if $G_1$ and $G_2$ are separated, we can just apply this algorithm.
Consequences

So we can assume that $d_{TV}(G_1, G_2) \leq 1 - \epsilon^c$, and $X = (G_1 + G_2)/2$. Letting $\Sigma = \text{Cov}(X)$, we have for some small $c' > 0$:

1. $\|\Sigma^{-1/2}(\Sigma_i - \Sigma)\Sigma^{-1/2}\|_F < (1/\epsilon)^c'$.
2. $\Sigma_i > \epsilon^{c'} \Sigma$.

This implies that everything is operating at the same scale. It says that after normalizing $X$, it suffices to learn $\mu_i, \Sigma_i$ to error $\text{poly}(\epsilon)$ in $L^2$, Frobenius norm, respectively.
Normalization

The next step is to normalize $X$. For this, we need to compute $\hat{\Sigma}, \hat{\mu}$ so that

$$
\| \Sigma^{-1/2} (\hat{\Sigma} - \Sigma) \Sigma^{-1/2} \|_F, \quad \| \Sigma^{-1/2} (\hat{\mu} - \mu) \|_2
$$

are small.

We can do this by generalizing the technique for a single Gaussian.
Given a Gaussian distribution $X$ with mean $\mu$ and covariance $\Sigma$, there exists an efficient algorithm that given $\epsilon$-corrupted samples from $X$ computes $\hat{\mu}$ and $\hat{\Sigma}$ so that with high probability

$$\|\Sigma^{-1/2}(\hat{\Sigma} - \Sigma)\Sigma^{-1/2}\|_F, \|\Sigma^{-1/2}(\hat{\mu} - \mu)\|_2 = \tilde{O}(\epsilon).$$

Fortunately, this result doesn’t require too many properties of the Gaussian distribution. It can be straightforwardly generalized.
General Normalization

**Theorem**

Given an arbitrary distribution \( X \) with mean \( \mu \) and covariance \( \Sigma \), so that for any matrix \( A \),

\[
\text{Var}( (X - \mu)^T A (X - \mu) ) \leq \sigma^2 \| \Sigma^{1/2} A \Sigma^{1/2} \|_F^2,
\]

there exists an efficient algorithm that given \( \epsilon \)-corrupted samples from \( X \) computes \( \hat{\mu} \) and \( \hat{\Sigma} \) so that with high probability

\[
\| \Sigma^{-1/2} (\hat{\Sigma} - \Sigma) \Sigma^{-1/2} \|_F, \quad \| \Sigma^{-1/2} (\hat{\mu} - \mu) \|_2 = O(\sigma \sqrt{\epsilon}).
\]

- This is proved using the same technique.
- The extra assumption applies with \( \sigma = (1/\epsilon)^O(c) \) for mixtures of non-separated Gaussians.
- This lets us normalize our distribution so that (up to small error) we can assume \( \text{Cov}(X) = I \) and \( \mathbb{E}[X] = 0 \).
Robust Moments

What do we do next? [Moitra-Valiant '10] suggests that we should compute moments. How do we do this robustly?

**Theorem ([DKKLMS17])**

Let $X$ be a distribution with $\text{Cov}(X) \leq \sigma^2 I$. There exists an efficient algorithm that given $\epsilon$-corrupted samples from $X$ computes a $\hat{\mu}$ so that with high probability

$$\|\hat{\mu} - \mathbb{E}[X]\|_2 = O(\sigma \sqrt{\epsilon}).$$

Apply to higher tensors of $X$. 
Moments

What are the moments of a Gaussian?

**Lemma**

Let $G = N(\mu, \Sigma)$, then

$$
\mathbb{E}[G^{\otimes m}]_{i_1...i_m} = \sum_{\text{Partitions } P \text{ of } [m] \text{ into sets of size 1 and 2}} \bigotimes_{\{a,b\} \in P} \Sigma_{i_a,i_b} \bigotimes_{\{c\} \in P} \mu_{i_c}.
$$

Problem: Covariance of $G^{\otimes m}$ is *not* bounded for $m \geq 3$. 
The problem is that the monomial basis doesn’t work so well for Gaussians. Let \( h_m(X) \) be the tensor whose entries are the (properly normalized) degree-\( m \) Hermite polynomials in \( X \).

**Lemma**

Let \( G = N(\mu, \Sigma) \), then

\[
\mathbb{E}[h_m(G)]_{i_1...i_m} = \sum_{\text{Partitions } P \text{ of } [m] \text{ into sets of size 1 and 2}} \bigotimes_{\{a,b\} \in P} (\Sigma_{i_a, i_b} - I) \bigotimes_{\{c\} \in P} \mu_{i_c}.
\]

Furthermore, \( \text{Cov}(h_m(X)) \) is bounded in terms of \( \|\Sigma_i - I\|_F \) and \( \|\mu_i\|_2 \). Thus, we can compute these robustly.
Some Computations

So \( X = (G_1 + G_2)/2 \) has mean 0 and covariance \( I \). Can write as

\[
G_1 = N(\mu, I + \Sigma - \mu\mu^T), \quad G_2 = N(-\mu, I - \Sigma - \mu\mu^T).
\]

Need to estimate \( \Sigma, \mu \) to \( \text{poly}(\epsilon) \) error in Frobenius/\( L^2 \) norm.

Can robustly compute Hermite moments

\[
\mathbb{E}[h_4(X)] = \text{Sym}(3\Sigma \otimes^2 - 2\mu \otimes^4),
\]

\[
\mathbb{E}[h_6(X)] = 16\mu \otimes^6.
\]
Estimating $\mu$

We have an approximation $T_6 \approx \mathbb{E}[h_6(X)] = 16\mu \otimes 6$. We note that $\mu$ is the principal (only) singular vector of the flattening of $\mathbb{E}[h_6(X)]$. Taking an appropriate multiple of the top singular vector of the flattening of $T_6$ gives an appropriate approximation to $\mu$.

Given this and an approximation to $\mathbb{E}[h_4(X)] = \text{Sym}(3\Sigma \otimes^2 - 2\mu \otimes^4)$, we compute $T \approx \text{Sym}(3\Sigma \otimes^2)$, we would like to compute $\Sigma$ from this.

Note, if we had $T \approx \Sigma \otimes^2$, we could apply the same technique. However the symmetrization gives us extraneous terms.
Recall $T \approx \text{Sym}(3\Sigma \otimes 2)$.

Consider for random $x, y$

$$T(x, y, -, -) \approx (x^T \Sigma y)\Sigma + (\Sigma x)(\Sigma y)^T + (\Sigma y)(\Sigma x)^T.$$ 

This is a multiple of $\Sigma$ plus low rank terms. If only we had a way to get rid of the low rank bits.
Low Rank Bits

\[ T(x, y, -, -) \approx (x^T \Sigma y) \Sigma + (\Sigma x)(\Sigma y)^T + (\Sigma y)(\Sigma x)^T. \]
\[ T(z, w, -, -) \approx (z^T \Sigma w) \Sigma + (\Sigma z)(\Sigma w)^T + (\Sigma w)(\Sigma z)^T. \]

Taking a linear combination and getting lucky, the \( \Sigma \) terms cancel and we get a low rank matrix. If \( \Sigma x, \Sigma y, \Sigma z, \Sigma w \) are linearly independent (which is probably true unless \( \Sigma \) is close to low rank), we can guess \( \Sigma x \) and \( \Sigma y \) from this span. Using this, we can approximate \( \Sigma \).
If $\Sigma$ itself is low rank, the above fails. However, the span of $\Sigma$ is the span of the flattening of $\text{Sym}(3\Sigma \otimes 2) \approx T$. We can:

- Flatten $T$ and let $V$ be the span of the top few singular vectors.
- Guess an approximation to $\Sigma$ within $V \otimes V$ (which is of bounded dimension).
Summary

- Guess if components are separated
  - No → Normalize Distribution
    - Approximate 4th & 6th Hermite Moments
      - Flatten 6th moment. Use principal eigenvector to learn μ.
  - Yes → Run DHKK/BK20
    - Flatten T and let V be span of principal eigenvectors. Guess Σ in V x V.
      - Yes → Guess if Σ is low rank.
        - No → Compute approximation T to Sym(3Σ^2)
          - Yes → Compute T(x_1, y_1, r, r) - T(w, z_1, r, r). Flatten and guess Σx, Σy in span. Use those and T(x, y, r, r) to approximate Σ.
            - Repeat several times and feed into robust tournament.
          - Answer

Kane (UCSD)  Robust GMMs  January, 2021
This provides the first efficient algorithm for robustly learning mixtures of arbitrary Gaussians.

Followup work by Bakshi-Diakonikolas-Jia-K-Kothari-Vempala and by Liu-Moitra have generalizations to mixtures of $k$ Gaussians.


Ilias Diakonikolas, Samuel B. Hopkins, Daniel Kane, Sushrut Karmalkar, Robustly Learning any Clusterable Mixture of Gaussians, FOCS 2020.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, Alistair Stewart, Robust Estimators in High Dimensions, without the Computational Intractability, Foundations Of Computer


