Computational Efficiency
and
Robust Statistics

Ilias Diakonikolas (USC)

(joint work with G. Kamath, D. Kane, J. Li, A. Moitra, and A. Stewart)
ROBUST PARAMETER ESTIMATION

Given corrupted samples from a Gaussian:

\[ \mathcal{N}(\mu, \sigma^2) \]

ideal model + noise = observed model

can we accurately estimate its parameters?
How do we constrain the noise?
How do we constrain the noise?

Equivalently:

$L_1$-norm of noise at most $O(\epsilon)$
How do we constrain the noise?

Equivalently:

$L_1$-norm of noise at most $O(\varepsilon)$

$L_1$-distance between observed and ideal models at most $O(\varepsilon)$
How do we constrain the noise?

Equivalently:

$L_1$-norm of noise at most $O(\epsilon)$

$L_1$-distance between observed and ideal models at most $O(\epsilon)$

This generalizes Huber’s Contamination Model:

$\text{(observed model)} = (1-\epsilon) \text{(ideal model)} + \epsilon \text{(noise)}$
In what norm do we want the parameters to be close?
In what norm do we want the parameters to be close?

**Definition:** The total variation distance between two distributions with pdfs \( f(x) \) and \( g(x) \) is

\[
d_{TV}(f(x), g(x)) \triangleq \frac{1}{2} \int_{-\infty}^{\infty} |f(x) - g(x)| \, dx
\]
In what norm do we want the parameters to be close?

**Definition:** The total variation distance between two distributions with pdfs $f(x)$ and $g(x)$ is

$$d_{TV}(f(x), g(x)) \triangleq \frac{1}{2} \int_{-\infty}^{\infty} \left| f(x) - g(x) \right| \, dx$$

From the bound on the $L_1$-norm of the noise, we have:

$$d_{TV}(\text{ideal}, \text{observed}) \leq O(\epsilon)$$
In what norm do we want the parameters to be close?

**Definition:** The total variation distance between two distributions with pdfs $f(x)$ and $g(x)$ is

$$d_{TV}(f(x), g(x)) \triangleq \frac{1}{2} \int_{-\infty}^{\infty} \left| f(x) - g(x) \right| \, dx$$

Find a 1-D Gaussian that satisfies

$$d_{TV}\left(\text{estimate}, \text{observed}\right) \leq O(\epsilon)$$
OUTLINE

Part I: Introduction

• Robust Estimation
• Our Results

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• Filtering
• Unknown Covariance

Part III: Experiments and Extensions
CONTAMINATION MODEL

**Definition (Contamination Model)**
Let $\mathcal{F}$ be a family of high-dimensional distributions. We say that a set of $N$ samples is $\epsilon$-corrupted from $\mathcal{F}$ if it is generated as follows:

- $N$ samples are drawn from an unknown $F \in \mathcal{F}$
- An omniscient adversary inspects these samples and changes arbitrarily an $\epsilon$-fraction of them.
**Definition (Contamination Model)**

Let $\mathcal{F}$ be a family of high-dimensional distributions. We say that a set of $N$ samples is $\varepsilon$-corrupted from $\mathcal{F}$ if it is generated as follows:

- $N$ samples are drawn from an unknown $F \in \mathcal{F}$
- An omniscient adversary inspects these samples and changes arbitrarily an $\varepsilon$-fraction of them.

**Remarks:**

- Adversary is adaptive.
**Definition (Contamination Model)**

Let $\mathcal{F}$ be a family of high-dimensional distributions. We say that a set of $N$ samples is $\epsilon$-corrupted from $\mathcal{F}$ if it is generated as follows:

- $N$ samples are drawn from an unknown $F \in \mathcal{F}$
- An omniscient adversary inspects these samples and changes arbitrarily an $\epsilon$-fraction of them.

**Remarks:**

- Adversary is adaptive.
- Non-adaptive adversary: samples from a distribution such that $d_{TV}(F', \mathcal{F}) \leq \epsilon$. 

CONTAMINATION MODEL
CONTAMINATION MODEL

Definition (Contamination Model)
Let $\mathcal{F}$ be a family of high-dimensional distributions. We say that a set of $N$ samples is $\epsilon$-corrupted from $\mathcal{F}$ if it is generated as follows:

- $N$ samples are drawn from an unknown $F \in \mathcal{F}$
- An omniscient adversary inspects these samples and changes arbitrarily an $\epsilon$-fraction of them.

Remarks:
- Adversary is adaptive.
- Non-adaptive adversary: samples from a distribution such that $d_{TV}(F', \mathcal{F}) \leq \epsilon$.
- Adaptive Huber’s model: adversary adds corrupted points, but not allowed to remove good samples.
**Main Problem:** Given an $\epsilon$ - corrupted set of samples from a $d$-dimensional unknown Gaussian

$$\mathcal{N}(\mu, \Sigma)$$

efficiently compute parameters $\hat{\mu}, \hat{\Sigma}$ that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) = O(\epsilon)$$
**Main Problem:** Given an $\epsilon$-corrupted set of samples from a $d$-dimensional unknown Gaussian

$$\mathcal{N}(\mu, \Sigma)$$

efficiently compute parameters $\hat{\mu}, \hat{\Sigma}$ that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) = O(\epsilon)$$

---

**Special Cases:**

1. Unknown mean $\mathcal{N}(\mu, I)$

2. Unknown covariance $\mathcal{N}(0, \Sigma)$
Theorem [Diakonikolas, Kamath, Kane, Li, Moitra, Stewart ‘16]:
There is an algorithm with the following behavior: Given $\epsilon > 0$ and an $\epsilon$ - corrupted set of $N$ samples from a $d$-dimensional Gaussian, $\mathcal{N}(\mu, \Sigma)$, the algorithm runs in $\text{poly}(N, d, 1/\epsilon)$ time and finds parameters $\hat{\mu}, \hat{\Sigma}$ that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) = O(\epsilon \log(1/\epsilon)) + \tilde{O}\left(\frac{d}{\sqrt{N}}\right).$$
Theorem [Diakonikolas, Kamath, Kane, Li, Moitra, Stewart ‘16]:
There is an algorithm with the following behavior: Given $\epsilon > 0$ and an $\epsilon$ - corrupted set of $N$ samples from a $d$-dimensional Gaussian, $\mathcal{N}(\mu, \Sigma)$, the algorithm runs in $\text{poly}(N, d, 1/\epsilon)$ time and finds parameters $\hat{\mu}, \hat{\Sigma}$ that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) = O(\epsilon \log(1/\epsilon)) + \tilde{O}\left(\frac{d}{\sqrt{N}}\right).$$

Corollary: For $N = \tilde{O}(d^2/\epsilon^2)$, we obtain a $\text{poly}(d/\epsilon)$ time algorithm with total variation distance error $O(\epsilon \log(1/\epsilon))$. 

MAIN RESULT
MAIN RESULT

**Theorem [Diakonikolas, Kamath, Kane, Li, Moitra, Stewart ‘16]:**
There is an algorithm with the following behavior: Given $\epsilon > 0$ and an $\epsilon$-corrupted set of $N$ samples from a $d$-dimensional Gaussian, $\mathcal{N}(\mu, \Sigma)$, the algorithm runs in $\text{poly}(N, d, 1/\epsilon)$ time and finds parameters $\hat{\mu}, \hat{\Sigma}$ that satisfy

$$d_{\text{TV}}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) = O(\epsilon \log(1/\epsilon)) + \tilde{O}\left(\frac{d}{\sqrt{N}}\right).$$

**Corollary:** For $N = \tilde{O}(d^2/\epsilon^2)$, we obtain a $\text{poly}(d/\epsilon)$ time algorithm with total variation distance error $O(\epsilon \log(1/\epsilon))$.

Near-sample optimal, even without noise!
A GENERAL TECHNIQUE

Robust estimation in high-dimensions:

• **Step #1:** Calculate a naïve estimator
  (requires appropriate parameter distance)

• **Step #2:** Detect when the naïve estimator is compromised

• **Step #3:** If it is, detect and remove outliers
  (requires concentration bound)
OUTLINE

Part I: Introduction

• Robust Estimation
• Our Results

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• Filtering
• Unknown Covariance

Part III: Experiments and Extensions
OUTLINE

Part I: Introduction
  • Robust Estimation
  • Our Results

Part II: Agnostically Learning a Gaussian
  • Parameter Distance
  • Detecting When an Estimator is Compromised
  • Filtering
  • Unknown Covariance

Part III: Experiments and Extensions
PARAMETER DISTANCE

**Step #1:** Find an appropriate parameter distance for Gaussians

**Basic Fact:**

\[
d_{TV}(\mathcal{N}(\mu, I), \mathcal{N} (\hat{\mu}, I)) \leq \frac{||\mu - \hat{\mu}\||_2}{2}
\]
PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

Basic Fact:

$$d_{TV}(N(\mu, I), N(\hat{\mu}, I)) \leq \frac{\|\mu - \hat{\mu}\|_2^2}{2}$$

Corollary: In the unknown mean case, it suffices to output an estimate $\hat{\mu}$ for the true mean $\mu$ satisfying

$$\|\mu - \hat{\mu}\|_2 = \tilde{O}(\epsilon) .$$

Our new goal is to be close in Euclidean distance.
OUTLINE

Part I: Introduction
  • Robust Estimation
  • Our Results

Part II: Agnostically Learning a Gaussian
  • Parameter Distance
  • Detecting When an Estimator is Compromised
  • Filtering
  • Unknown Covariance

Part III: Experiments and Extensions
OUTLINE

Part I: Introduction

• Robust Estimation
• Our Results

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• Filtering
• Unknown Covariance

Part III: Experiments and Extensions
DETECTING CORRUPTIONS

**Step #2:** Detect when the naïve estimator has been compromised
DETECTING CORRUPTIONS

Step #2: Detect when the naïve estimator has been compromised

\[ \hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} X_i \]

- \( \bullet \) = uncorrupted
- \( \bullet \) = corrupted
DETECTING CORRUPTIONS

Step #2: Detect when the naïve estimator has been compromised

There is a direction of large (> 1) variance

\[ \hat{\mu} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i \]
**Key Lemma:** If $X_1, X_2, ..., X_N$ come from a distribution that is $\varepsilon$-close to $\mathcal{N}(\mu, I)$ and $N \geq 10(d + \log 1/\delta)/\varepsilon^2$ then for

\[
(1) \quad \hat{\mu} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i \quad \text{ (2) } \quad \hat{\Sigma} \triangleq \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{\mu})(X_i - \hat{\mu})^T
\]

with probability at least $1-\delta$

\[
\|\mu - \hat{\mu}\|_2 \geq C\varepsilon \sqrt{\log 1/\varepsilon} \quad \Rightarrow \quad \|\hat{\Sigma} - I\|_2 \geq C'\varepsilon \log 1/\varepsilon
\]
**Key Lemma:** If $X_1, X_2, \ldots, X_N$ come from a distribution that is $\epsilon$-close to $\mathcal{N}(\mu, I)$ and $N \geq 10(d + \log 1/\delta)/\epsilon^2$ then for

\[(1)\]  $\hat{\mu} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i$ \hspace{1cm} \[(2)\]  $\hat{\Sigma} \triangleq \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{\mu})(X_i - \hat{\mu})^T$

with probability at least $1-\delta$

$$\|\mu - \hat{\mu}\|_2 \geq C\epsilon \sqrt{\log 1/\epsilon} \quad \Rightarrow \quad \|\hat{\Sigma} - I\|_2 \geq C'\epsilon \log 1/\epsilon$$

**Take-away:** An adversary needs to mess up the second moment in order to corrupt the first moment
OUTLINE

Part I: Introduction
  • Robust Estimation
  • Our Results

Part II: Agnostically Learning a Gaussian
  • Parameter Distance
  • Detecting When an Estimator is Compromised
  • Filtering
  • Unknown Covariance

Part III: Experiments and Extensions
OUTLINE

Part I: Introduction

- Robust Estimation
- Our Results

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- Filtering
- Unknown Covariance

Part III: Experiments and Extensions
FILTERING ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

\[ \| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon \]

Let \( v \) be the direction of maximum variance.
FILTERING ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

$$
\| \hat{\Sigma} - I \|_2 \geq C' \varepsilon \log \frac{1}{\varepsilon}
$$

Let $v$ be the direction of maximum variance.
FILTERING ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

\[ \| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon \]

Let \( v \) be the direction of maximum variance.
FILTERING ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

$$\|\hat{\Sigma} - I\|_2 \geq C' \epsilon \log 1/\epsilon$$

Let $v$ be the direction of maximum variance.
FILTERING ALGORITHM

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

\[ \| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon \]

Let \( \nu \) be the direction of maximum variance.
FILTERING ALGORITHM

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

$$\|\hat{\Sigma} - I\|_2 \geq C' \epsilon \log 1/\epsilon$$

Let $\nu$ be the direction of maximum variance.

• Project all the points on the direction of $\nu$. 
FILTERING ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

\[ \| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon \]

Let \( \nu \) be the direction of maximum variance.

- Project all the points on the direction of \( \nu \).
- Find a threshold \( T \) such that

\[
\Pr_{x \sim u_S} [ | \nu \cdot x - \text{median}(\nu \cdot x) | > T ] \geq 3e^{-T^2/2}
\]
FILTERING ALGORITHM

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

\[ \|\hat{\Sigma} - I\|_2 \geq C' \epsilon \log 1/\epsilon \]

Let \( \nu \) be the direction of maximum variance.

• Project all the points on the direction of \( \nu \).
• Find a threshold \( T \) such that

\[ \text{Pr}_{x \sim u_S} \left[ |\nu \cdot x - \text{median}(\nu \cdot x)| > T \right] \geq 3e^{-T^2/2} \]
• Throw away all points such that

\[ |\nu \cdot x - \text{median}(\nu \cdot x)| > T \]
FILTERING ALGORITHM

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

$$\|\hat{\Sigma} - I\|_2 \geq C' \epsilon \log 1/\epsilon$$

Let $v$ be the direction of maximum variance.

- Project all the points on the direction of $v$.
- Find a threshold $T$ such that
  $$\Pr_{x \sim u,S} [|v \cdot x - \text{median}(v \cdot x)| > T] \geq 3e^{-T^2/2}$$
- Throw away all points such that
  $$|v \cdot x - \text{median}(v \cdot x)| > T$$
- Iterate on new dataset.
OUR ALGORITHM(S)

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

\[ \| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon \]

We can throw out more corrupted than uncorrupted points

If we continue too long, we’d have no corrupted points left!
OUR ALGORITHM(S)

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

$$\| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log 1/\epsilon$$

We can throw out more corrupted than uncorrupted points

If we continue too long, we’d have no corrupted points left!

Eventually we find (certifiably) good parameters
**OUR ALGORITHM(S)**

**Step #3:** Either find good parameters, or remove many outliers

**Filtering Approach:** Suppose that:

$$\|\hat{\Sigma} - I\|_2 \geq C' \epsilon \log 1/\epsilon$$

We can throw out more corrupted than uncorrupted points

If we continue too long, we’d have no corrupted points left!

Eventually we find (certifiably) good parameters

**Running Time:** $\tilde{O}(Nd^2)$  
**Sample Complexity:** $N = \tilde{O}(d/\epsilon^2)$
OUR ALGORITHM(S)

Step #3: Either find good parameters, or remove many outliers

Filtering Approach: Suppose that:

$$\| \hat{\Sigma} - I \|_2 \geq C' \epsilon \log \frac{1}{\epsilon}$$

We can throw out more corrupted than uncorrupted points

If we continue too long, we’d have no corrupted points left!

Eventually we find (certifiably) good parameters

Running Time: $\tilde{O}(Nd^2)$  
Sample Complexity: $N = \tilde{O}(d/\epsilon^2)$

Sample optimal!
GENERALITY OF ROBUST MEAN ESTIMATION

- Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee: $$\|\mu - \hat{\mu}\|_2 = O(\epsilon \sqrt{\log(1/\epsilon)})$$
GENERALITY OF ROBUST MEAN ESTIMATION

• Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee: \[ \| \mu - \hat{\mu} \|_2 = O(\epsilon \sqrt{\log(1/\epsilon)}) \]

• Algorithm works under weaker concentration assumptions with appropriate guarantees.
GENERALITY OF ROBUST MEAN ESTIMATION

• Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee: \[ \| \mu - \hat{\mu} \|_2 = O(\varepsilon \sqrt{\log(1/\varepsilon)}) \]

• Algorithm works under weaker concentration assumptions with appropriate guarantees.
  E.g.,
  Under 2\textsuperscript{nd} moment assumption: \( O(\sqrt{\varepsilon}) \)
GENERALITY OF ROBUST MEAN ESTIMATION

• Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee: \[ \| \mu - \hat{\mu} \|_2 = O(\epsilon \sqrt{\log(1/\epsilon)}) \]

• Algorithm works under weaker concentration assumptions with appropriate guarantees.
  E.g.,
  Under 2\textsuperscript{nd} moment assumption: \( O(\sqrt{\epsilon}) \)
  Under 4\textsuperscript{th} moment assumption: \( O(\epsilon^{3/4}) \)
GENERALITY OF ROBUST MEAN ESTIMATION

• Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee: $\|\mu - \hat{\mu}\|_2 = O(\epsilon \sqrt{\log(1/\epsilon)})$

• Algorithm works under weaker concentration assumptions with appropriate guarantees.
  E.g.,
  Under 2\textsuperscript{nd} moment assumption: $O(\sqrt{\epsilon})$
  Under 4\textsuperscript{th} moment assumption: $O(\epsilon^{3/4})$
  Under sub-gaussian assumption $O(\epsilon \sqrt{\log(1/\epsilon)})$
  ...

GENERALITY OF ROBUST MEAN ESTIMATION

• Focus of initial version was on specific distribution families (e.g., Gaussian, discrete product distributions).

Error guarantee:  \[ \| \mu - \hat{\mu} \|_2 = O(\epsilon \sqrt{\log(1/\epsilon)}) \]

• Algorithm works under weaker concentration assumptions with appropriate guarantees.
  E.g.,
  Under 2\textsuperscript{nd} moment assumption:  \[ O(\sqrt{\epsilon}) \]
  Under 4\textsuperscript{th} moment assumption:  \[ O(\epsilon^{3/4}) \]
  Under sub-gaussian assumption  \[ O(\epsilon \sqrt{\log(1/\epsilon)}) \]
  ...

• Sample complexity near-optimal for all these cases.
OUTLINE

Part I: Introduction

• Robust Estimation in One-dimension
• Robustness vs. Hardness in High-dimensions
• Our Results

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• Filtering
• Unknown Covariance

Part III: Experiments and Extensions
OUTLINE

Part I: Introduction
  • Robust Estimation in One-dimension
  • Robustness vs. Hardness in High-dimensions
  • Our Results

Part II: Agnostically Learning a Gaussian
  • Parameter Distance
  • Detecting When an Estimator is Compromised
  • Filtering
  • Unknown Covariance

Part III: Experiments and Extensions
A GENERAL TECHNIQUE

Robust estimation in high-dimensions:

| • **Step #1:** Calculate a naïve estimator  
  (requires appropriate parameter distance) |
| • **Step #2:** Detect when the naïve estimator is compromised |
| • **Step #3:** If it is, detect and remove outliers  
  (requires concentration bound) |
PARAMETER DISTANCE

**Step #1:** Find an appropriate parameter distance for Gaussians
PARAMETER DISTANCE

**Step #1:** Find an appropriate parameter distance for Gaussians

**Another Basic Fact:**

\[
(2) \quad d_{TV}(\mathcal{N}(0, \Sigma), \mathcal{N}(0, \hat{\Sigma})) \leq O(\|I - \hat{\Sigma}^{-1/2} \Sigma \hat{\Sigma}^{-1/2}\|_F)
\]
PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

Another Basic Fact:

\[(2) \quad d_{TV}(\mathcal{N}(0, \Sigma), \mathcal{N}(0, \hat{\Sigma})) \leq O(\| I - \hat{\Sigma}^{-1/2} \Sigma \hat{\Sigma}^{-1/2} \|_F) \]

Our new goal is to find an estimate that satisfies:

\[\| I - \hat{\Sigma}^{-1/2} \Sigma \hat{\Sigma}^{-1/2} \|_F \leq \tilde{O}(\epsilon)\]
UNKNOWN COVARIANCE

What if we are given samples from $\mathcal{N}(0, \Sigma)$?
UNKNOWN COVARIANCE

What if we are given samples from $\mathcal{N}(0, \Sigma)$?

How do we detect if the naïve estimator is compromised?

$$\hat{\Sigma} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i X_i^T$$
UNKNOWN COVARIANCE

What if we are given samples from $\mathcal{N}(0, \Sigma)$?

How do we detect if the naïve estimator is compromised?

$$\hat{\Sigma} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i X_i^T$$

**Key Fact:** Let $X_i \sim \mathcal{N}(0, \Sigma)$ and $M = \mathbb{E}[(X_i \otimes X_i)(X_i \otimes X_i)^T]$

Then restricted to flattenings of $d \times d$ symmetric matrices

$$M = 2\Sigma \otimes^2 + \begin{pmatrix} \Sigma^b \end{pmatrix} \begin{pmatrix} \Sigma^b \end{pmatrix}^T$$
Key Idea: Transform the data, look for restricted large eigenvalues
**Key Idea:** Transform the data, look for restricted large eigenvalues

\[ Y_i \triangleq (\hat{\Sigma})^{-1/2} X_i \]
**Key Idea:** Transform the data, look for restricted large eigenvalues

\[ Y_i \triangleq (\Sigma)^{-1/2} X_i \]

If \( \Sigma \) were the true covariance, we would have \( Y_i \sim N(0, I) \) for inliers
**Key Idea:** Transform the data, look for restricted large eigenvalues

\[ Y_i \triangleq (\hat{\Sigma})^{-1/2}X_i \]

If \( \hat{\Sigma} \) were the true covariance, we would have \( Y_i \sim N(0, I) \) for inliers, in which case:

\[
\frac{1}{N} \sum_{i=1}^{N} (Y_i \otimes Y_i) (Y_i \otimes Y_i)^T - 2I - (I^b)(I^b)^T
\]

would have small eigenvalues
**Key Idea:** Transform the data, look for restricted large eigenvalues

\[ Y_i \triangleq (\Sigma)^{-1/2} X_i \]

If \( \Sigma \) were the true covariance, we would have \( Y_i \sim N(0, I) \) for inliers, in which case:

\[
\frac{1}{N} \sum_{i=1}^{N} (Y_i \otimes Y_i) (Y_i \otimes Y_i)^T - 2I - (I^b)(I^b)^T
\]

would have small eigenvalues

**Take-away:** An adversary needs to mess up the (restricted) fourth moment in order to corrupt the second moment
OUTLINE

Part I: Introduction
  • Robust Estimation
  • Our Results

Part II: Agnostically Learning a Gaussian
  • Parameter Distance
  • Detecting When an Estimator is Compromised
  • Filtering
  • Unknown Covariance

Part III: Experiments and Extensions
OUTLINE

Part I: Introduction

- Robust Estimation
- Our Results

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- Filtering
- Unknown Covariance

Part III: Experiments and Extensions
FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers
FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

**Binary Product Distributions:**

\[ d_{TV}(\Pi, \hat{\Pi}) \leq C \sqrt{\epsilon \log \frac{1}{\epsilon}} \]
FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

**Binary Product Distributions:**

\[ d_{TV}(\Pi, \hat{\Pi}) \leq C \sqrt{\epsilon \log 1/\epsilon} \]

**Mixtures of Two c-Balanced Binary Product Distributions:**

\[ d_{TV}(\Pi, \hat{\Pi}) \leq C \epsilon^{1/6} \]
FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

**Binary Product Distributions:**

\[ d_{TV}(\Pi, \hat{\Pi}) \leq C \sqrt{\varepsilon \log 1/\varepsilon} \]

**Mixtures of Two c-Balanced Binary Product Distributions:**

\[ d_{TV}(\Pi, \hat{\Pi}) \leq C \varepsilon^{1/6} \]

**Mixtures of k Spherical Gaussians:**

\[ d_{TV}(\mathcal{M}, \hat{\mathcal{M}}) \leq C \text{ poly}(k) \sqrt{\varepsilon \log 1/\varepsilon} \]
PRELIMINARY EXPERIMENTS (I)

A comparison of error rate on synthetic data (unknown mean):

![Graph showing comparison of Geometric Median, RANSAC, and Our Algorithm]
PRELIMINARY EXPERIMENTS (II)

A comparison of error rate on synthetic data (unknown covariance):
SUMMARY AND CONCLUSIONS

Other estimators have curse-of-dimensionality, ours doesn’t!
SUMMARY AND CONCLUSIONS

Other estimators have **curse-of-dimensionality**, ours doesn’t!

**Next Steps:**

- Optimal Guarantees for Gaussian?
  - D-Kamath-Kane-Li-Moitra-Stewart’17: **YES!** (additive model)
  - D-Kane-Stewart’17: **Statistical Query Barrier** (general model)

- More general probabilistic models?
  - [D-Kane-Stewart’17] **Known Structure Bayes Nets**
Summary:

• Nearly optimal algorithm for agnostically learning a high-dimensional Gaussian
• General method using restricted eigenvalue problems
• Further applications to other “simple” models
• Is practical, robust statistics within reach?

Thanks! Any Questions?