Robustness Meets Algorithms

Ankur Moitra (MIT)

HALG 2018 Survey, June 5th
CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some class e.g. a 1-D Gaussian $\mathcal{N}(\mu, \sigma^2)$ can we accurately estimate its parameters?
CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some class e.g. a 1-D Gaussian

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

can we accurately estimate its parameters? Yes!
CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some class e.g. a 1-D Gaussian $\mathcal{N}(\mu, \sigma^2)$

can we accurately estimate its parameters? Yes!

**empirical mean:**

$$\frac{1}{N} \sum_{i=1}^{N} X_i \to \mu$$

**empirical variance:**

$$\frac{1}{N} \sum_{i=1}^{N} (X_i - \overline{X})^2 \to \sigma^2$$
The maximum likelihood estimator is asymptotically efficient (1910-1920)

R. A. Fisher
The **maximum likelihood estimator** is asymptotically efficient (1910-1920)

What about **errors** in the model itself? (1960)
What estimators behave well in a \textit{neighborhood} around the model?
What estimators behave well in a neighborhood around the model?

Let’s study a simple one-dimensional example....
ROBUST PARAMETER ESTIMATION

Given corrupted samples from a 1-D Gaussian:

\[ N(\mu, \sigma^2) + \text{noise} = \text{observed model} \]

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 $O(\varepsilon)$
How do we constrain the noise?

Equivalently:

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

Arbitrarily corrupt $O(\epsilon)$-fraction of samples (in expectation)
How do we constrain the noise?

Equivalently:

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

Arbitrarily corrupt $O(\varepsilon)$-fraction of samples (in expectation)

This generalizes **Huber’s Contamination Model**: An adversary can add an $\varepsilon$-fraction of samples
How do we constrain the noise?

Equivalently:

L₁-norm of noise at most $O(\varepsilon)$

Arbitrarily corrupt $O(\varepsilon)$-fraction of samples (in expectation)

This generalizes **Huber’s Contamination Model**: An adversary can add an $\varepsilon$-fraction of samples

**Outliers**: Points adversary has corrupted, **Inliers**: Points he hasn’t
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

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

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

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

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

**Goal:** Find a 1-D Gaussian that satisfies

$$d_{TV}(\text{estimate}, \text{ideal}) \leq O(\epsilon)$$
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

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

Equivalently, find a 1-D Gaussian that satisfies

\[
d_{TV}(\text{estimate}, \text{observed}) \leq O(\epsilon)
\]
Do the empirical mean and empirical variance work?
Do the empirical mean and empirical variance work?

No!
Do the empirical mean and empirical variance work?

No!

ideal model + noise = observed model
Do the empirical mean and empirical variance work?

No!

A single corrupted sample can arbitrarily corrupt the estimates.
Do the empirical mean and empirical variance work?

No!

A single corrupted sample can arbitrarily corrupt the estimates

But the **median** and **median absolute deviation** do work
Do the empirical mean and empirical variance work?

No!

ideal model + noise = observed model

A single corrupted sample can arbitrarily corrupt the estimates

But the **median** and **median absolute deviation** do work

\[
\text{MAD} = \text{median}(|X_i - \text{median}(X_1, X_2, \ldots, X_n)|)
\]
**Fact [Folklore]:** Given samples from a distribution that are $\varepsilon$-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

$$d_{TV}(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)) \leq O(\varepsilon)$$

where $\hat{\mu} = \text{median}(X)$, $\hat{\sigma} = \frac{\text{MAD}}{\Phi^{-1}(3/4)}$
**Fact [Folklore]:** Given samples from a distribution that are $\epsilon$-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

$$d_{TV}(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)) \leq O(\epsilon)$$

where $\hat{\mu} = \text{median}(X)$, $\hat{\sigma} = \frac{\text{MAD}}{\Phi^{-1}(3/4)}$

Also called (properly) **agnostically learning** a 1-D Gaussian
Fact [Folklore]: Given samples from a distribution that are $\varepsilon$-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

$$d_{TV}(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)) \leq O(\varepsilon)$$

where $\hat{\mu} = \text{median}(X)$, $\hat{\sigma} = \frac{\text{MAD}}{\Phi^{-1}(3/4)}$

What about robust estimation in high-dimensions?
Fact [Folklore]: Given samples from a distribution that are $\epsilon$-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

$$d_{TV}(\mathcal{N}(\mu, \sigma^2), \mathcal{N}(\hat{\mu}, \hat{\sigma}^2)) \leq O(\epsilon)$$

where $\hat{\mu} = \text{median}(X)$, $\hat{\sigma} = \frac{\text{MAD}}{\Phi^{-1}(3/4)}$

What about robust estimation in high-dimensions?

e.g. microarrays with 10k genes
Part I: Introduction
• Robust Estimation in One-dimension
• Robustness vs. Hardness in High-dimensions
• Recent Results

Part II: Agnostically Learning a Gaussian
• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
**Main Problem:** Given samples from a distribution that are $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, give an efficient algorithm to find parameters that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq \tilde{O}(\epsilon)$$
Main Problem: Given samples from a distribution that are $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, give an efficient algorithm to find parameters that satisfy

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

Special Cases:

1. Unknown mean $\mathcal{N}(\mu, I)$
2. Unknown covariance $\mathcal{N}(0, \Sigma)$
## A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**A COMPENDIUM OF APPROACHES**

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$</td>
<td></td>
</tr>
<tr>
<td>Unknown Mean</td>
<td>Error Guarantee</td>
<td>Running Time</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$ ✓</td>
<td>NP-Hard X</td>
</tr>
<tr>
<td>Unknown Mean</td>
<td>Error Guarantee</td>
<td>Running Time</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$ ✓</td>
<td>NP-Hard ❌</td>
</tr>
<tr>
<td>Geometric Median</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$</td>
<td>NP-Hard</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>poly(d,N)</td>
<td></td>
</tr>
</tbody>
</table>

- **Tukey Median**
  - Error Guarantee: $O(\varepsilon)$ ✓
  - Running Time: NP-Hard X

- **Geometric Median**
  - Running Time: poly(d,N) ✓
# A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$ ✓</td>
<td>NP-Hard ×</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\varepsilon\sqrt{d})$ ×</td>
<td>poly(d,N) ✓</td>
</tr>
</tbody>
</table>
## A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$ ✓</td>
<td>NP-Hard X</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\varepsilon\sqrt{d})$ X</td>
<td>$\text{poly}(d,N)$ ✓</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\varepsilon)$ ✓</td>
<td>$N^{O(d)}$ X</td>
</tr>
<tr>
<td></td>
<td>Unknown Mean</td>
<td>Error Guarantee</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>O(\varepsilon)</td>
<td>✓</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>O(\varepsilon\sqrt{d})</td>
<td>X</td>
</tr>
<tr>
<td>Tournament</td>
<td>O(\varepsilon)</td>
<td>✓</td>
</tr>
<tr>
<td>Pruning</td>
<td>O(\varepsilon\sqrt{d})</td>
<td>X</td>
</tr>
</tbody>
</table>
# A Compendium of Approaches

<table>
<thead>
<tr>
<th>Unknown Mean</th>
<th>Error Guarantee</th>
<th>Running Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$ ✓</td>
<td>NP-Hard ×</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\varepsilon \sqrt{d})$ ×</td>
<td>$\text{poly}(d,N)$ ✓</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\varepsilon)$ ✓</td>
<td>$O^{O(d)}$ ×</td>
</tr>
<tr>
<td>Pruning</td>
<td>$O(\varepsilon \sqrt{d})$ ×</td>
<td>$O(dN)$ ✓</td>
</tr>
</tbody>
</table>
The Price of Robustness?

All known estimators are hard to compute or lose polynomial factors in the dimension.
The Price of Robustness?

All known estimators are **hard to compute** or lose **polynomial** factors in the dimension.

Equivalently: Computationally efficient estimators can only handle

$$\epsilon \leq \frac{1}{\sqrt{d}}$$

fraction of errors and get **non-trivial** (TV < 1) guarantees.
The Price of Robustness?

All known estimators are **hard to compute** or lose **polynomial** factors in the dimension.

Equivalently: Computationally efficient estimators can only handle

$$\epsilon \leq \frac{1}{100} \text{ for } d = 10,000$$

fraction of errors and get **non-trivial** (TV < 1) guarantees.
The Price of Robustness?

All known estimators are **hard to compute** or lose **polynomial** factors in the dimension.

Equivalently: Computationally efficient estimators can only handle

$$\epsilon \leq \frac{1}{100} \text{ for } d = 10,000$$

fraction of errors and get **non-trivial** (TV < 1) guarantees.

Is robust estimation algorithmically possible in high-dimensions?
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
RECENT RESULTS

Robust estimation is high-dimensions is algorithmically possible!

Theorem [Diakonikolas, Li, Kamath, Kane, Moitra, Stewart ‘16]: There is an algorithm when given $N = \tilde{O}(d^2/\epsilon^2)$ samples from a distribution that is $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq O(\epsilon \log^{3/2} 1/\epsilon)$$

Moreover the algorithm runs in time $\text{poly}(N, d)$
RECENT RESULTS

Robust estimation is high-dimensions is algorithmically possible!

**Theorem [Diakonikolas, Li, Kamath, Kane, Moitra, Stewart ‘16]:**
There is an algorithm when given $N = \tilde{O}(d^2/\epsilon^2)$ samples from a distribution that is $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

$$d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq O(\epsilon \log^{3/2} 1/\epsilon)$$

Moreover the algorithm runs in time $\text{poly}(N, d)$

**Alternatively:** Can approximate the Tukey median, etc, in beyond worst-case analysis sense
Independently and concurrently:

**Theorem [Lai, Rao, Vempala ‘16]:** There is an algorithm when given $N = \tilde{O}(d^2/\epsilon^2)$ samples from a distribution that is $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

$$\|\mu - \hat{\mu}\|_2 \leq C\epsilon^{1/2}\|\Sigma\|_2^{1/2} \log^{1/2} d$$

$$\|\Sigma - \hat{\Sigma}\|_F \leq C\epsilon^{1/2}\|\Sigma\|_2 \log^{1/2} d$$

Moreover the algorithm runs in time $\text{poly}(N, d)$
Independently and concurrently:

**Theorem [Lai, Rao, Vempala ‘16]:** There is an algorithm when given $N = \tilde{O}(d^2/\epsilon^2)$ samples from a distribution that is $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

\[
\|\mu - \hat{\mu}\|_2 \leq C\epsilon^{1/2} \|\Sigma\|_2^{1/2} \log^{1/2} d
\]

\[
\|\Sigma - \hat{\Sigma}\|_F \leq C\epsilon^{1/2} \|\Sigma\|_2 \log^{1/2} d
\]

Moreover the algorithm runs in time $\text{poly}(N, d)$

When the covariance is bounded, this translates to:

\[
d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma})) \leq \tilde{O}(\epsilon^{1/2})
\]
A GENERAL RECIPE

Robust estimation in high-dimensions:

- **Step #1:** Find an appropriate parameter distance
- **Step #2:** Detect when the naïve estimator has been compromised
- **Step #3:** Find good parameters, or make progress

**Filtering:** Fast and practical

**Convex Programming:** Better sample complexity
A GENERAL RECIPE

Robust estimation in high-dimensions:

• **Step #1:** Find an appropriate parameter distance

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

• **Step #3:** Find good parameters, or make progress

  **Filtering:** Fast and practical

  **Convex Programming:** Better sample complexity

Let’s see how this works for *unknown mean*...
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
PARAMETER DISTANCE

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

Step #1: Find an appropriate parameter distance for Gaussians

A 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

A Basic Fact:

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

This can be proven using Pinsker’s Inequality

\[ d_{TV}(f, g)^2 \leq \frac{1}{2} d_{KL}(f, g) \]

and the well-known formula for KL-divergence between Gaussians
PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

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

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

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

Corollary: If our estimate (in the unknown mean case) satisfies

\[
\|\mu - \hat{\mu}\|_2 \leq \tilde{O}(\varepsilon)
\]

then \(d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I)) \leq \tilde{O}(\varepsilon)\)
PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

\[(1) \quad d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I)) \leq \frac{\|\mu - \hat{\mu}\|_2}{2}\]

Corollary: If our estimate (in the unknown mean case) satisfies

\[\|\mu - \hat{\mu}\|_2 \leq O(\epsilon)\]

then \(d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I)) \leq O(\epsilon)\)

Our new goal is to be close in Euclidean distance
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
DETECTING CORRUPTIONS

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

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

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

- = uncorrupted
- = corrupted
DETECTING CORRUPTIONS

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

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

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

There is a direction of large (> 1) variance
Key Lemma: If $X_1, X_2, \ldots, X_N$ 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

$$\begin{align*}
(1) \quad \hat{\mu} & \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i \\
(2) \quad \hat{\Sigma} & \triangleq \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{\mu})(X_i - \hat{\mu})^T
\end{align*}$$

with probability at least $1-\delta$

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

\[
\begin{align*}
(1) \quad \hat{\mu} & \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i \\
(2) \quad \hat{\Sigma} & \triangleq \frac{1}{N} \sum_{i=1}^{N} (X_i - \hat{\mu})(X_i - \hat{\mu})^T
\end{align*}
\]

with probability at least 1-\( \delta \)

\[
\|\mu - \hat{\mu}\|_2 \geq C\varepsilon \sqrt{\log 1/\varepsilon} \quad \Rightarrow \quad \|\hat{\Sigma} - I\|_2 \geq C'\varepsilon \log 1/\varepsilon
\]

Take-away: An adversary needs to mess up the second moment in order to corrupt the first moment
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
A WIN-WIN ALGORITHM

**Step #3:** Either find good parameters, or remove many outliers
A WIN-WIN ALGORITHM

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

**Filtering Approach:** Suppose that:

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

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points:

where \( v \) is the direction of largest variance
A WIN-WIN ALGORITHM

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

**Filtering Approach:** Suppose that:

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

We can throw out more corrupted than uncorrupted points:

where \( v \) is the direction of largest variance, and \( T \) has a formula
A WIN-WIN ALGORITHM

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:

where \( v \) is the direction of largest variance, and \( T \) has a formula
A WIN-WIN ALGORITHM

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
A WIN-WIN ALGORITHM

**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!
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

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
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

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: $\tilde{O}(d^2/\epsilon^2)$
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

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: $\tilde{O}(d^2/\epsilon^2)$
Concentration of LTFs
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
A GENERAL RECIPE

Robust estimation in high-dimensions:

- **Step #1:** Find an appropriate parameter distance
- **Step #2:** Detect when the naïve estimator has been compromised
- **Step #3:** Find good parameters, or make progress

**Filtering:** Fast and practical

**Convex Programming:** Better sample complexity
A GENERAL RECIPE

Robust estimation in high-dimensions:

- **Step #1:** Find an appropriate parameter distance
- **Step #2:** Detect when the naïve estimator has been compromised
- **Step #3:** Find good parameters, or make progress

**Filtering:** Fast and practical

**Convex Programming:** Better sample complexity

How about for **unknown covariance**?
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:

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

Again, proven using Pinsker’s Inequality
PARAMETER DISTANCE

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

Another Basic Fact:

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

Again, proven using Pinsker’s Inequality

Our new goal is to find an estimate that satisfies:

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

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

**Another Basic Fact:**

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

Again, proven using Pinsker’s Inequality

Our new goal is to find an estimate that satisfies:

\[
\| I - \widehat{\Sigma}^{-1/2} \Sigma \widehat{\Sigma}^{-1/2} \|_F \leq \widetilde{O}(\epsilon)
\]

Distance seems strange, but it’s the right one to use to bound TV
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

$$M = 2\Sigma \otimes 2 + \left(\Sigma^b\right)\left(\Sigma^b\right)^T$$
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

$$M = 2\Sigma \otimes^2 + \left(\Sigma^b\right) \left(\Sigma^b\right)^T$$

Proof uses **Islerlis’s Theorem**
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

$$M = 2\Sigma \otimes 2 + \left(\Sigma^b\right) \left(\Sigma^b\right)^T$$

need to project out
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

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

If \( \hat{\Sigma} \) 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:

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

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

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

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

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

would have small restricted eigenvalues

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

Given samples that are $\varepsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$.
ASSEMBLING THE ALGORITHM

Given samples that are $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$

**Step #1:** Doubling trick $X_i - X_i' \sim \epsilon \mathcal{N}(0, 2\Sigma)$
ASSEMBLING THE ALGORITHM

Given samples that are $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$

**Step #1:** Doubling trick $X_i - X_i' \sim_{\epsilon} \mathcal{N}(0, 2\Sigma)$

Now use algorithm for **unknown covariance**
ASSEMBLING THE ALGORITHM

Given samples that are $\epsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$

**Step #1:** Doubling trick $X_i - X_i' \sim \epsilon \mathcal{N}(0, 2\Sigma)$

Now use algorithm for **unknown covariance**

**Step #2:** (Agnostic) isotropic position

$$\hat{\Sigma}^{-1/2} X_i \sim \epsilon \mathcal{N}(\hat{\Sigma}^{-1/2} \mu, I)$$
ASSEMBLING THE ALGORITHM

Given samples that are $\varepsilon$-close in total variation distance to a $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$

**Step #1:** Doubling trick $X_i - X_i' \sim \varepsilon \mathcal{N}(0, 2\Sigma)$

Now use algorithm for **unknown covariance**

**Step #2:** (Agnostic) isotropic position

$$\hat{\Sigma}^{-1/2} X_i \sim \varepsilon \mathcal{N}(\hat{\Sigma}^{-1/2} \mu, I)$$

right distance, in general case
ASSEMBLING THE ALGORITHM

Given samples that are ε-close in total variation distance to a d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$

**Step #1:** Doubling trick $X_i - X'_i \sim \epsilon \, \mathcal{N}(0, 2\Sigma)$

Now use algorithm for **unknown covariance**

**Step #2:** (Agnostic) isotropic position

$$\hat{\Sigma}^{-1/2} X_i \sim \epsilon \, \mathcal{N}(\hat{\Sigma}^{-1/2} \mu, I)$$

right distance, in general case

Now use algorithm for **unknown mean**
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
SYNTHEtic EXPERIMENTS

Error rates on synthetic data (unknown mean):

\[ \mathcal{N}(\mu, I) + 10\% \text{ noise} \]
SYNTHETIC EXPERIMENTS

Error rates on synthetic data (**unknown mean**):

![Graph showing error rates for different methods with varying dimensionality.](image-url)
SYNTHETIC EXPERIMENTS

Error rates on synthetic data (unknown covariance, isotropic):

\[ \mathcal{N}(0, \Sigma) + 10\% \text{ noise} \]

close to identity
SYNTHETIC EXPERIMENTS

Error rates on synthetic data (unknown covariance, isotropic):
SYNTHETIC EXPERIMENTS

Error rates on synthetic data (unknown covariance, anisotropic):

\[ N(0, \Sigma) + 10\% \text{ noise} \]

far from identity
SYNTHETIC EXPERIMENTS

Error rates on synthetic data (unknown covariance, anisotropic):

- Filtering
- LRVCov
- Sample covariance w/ noise
- Pruning
- RANSAC
REAL DATA EXPERIMENTS

Famous study of [Novembre et al. ‘08]: Take top two singular vectors of people x SNP matrix (POPRES)
REAL DATA EXPERIMENTS

Famous study of [Novembre et al. ‘08]: Take top two singular vectors of people x SNP matrix (POPRES)
REAL DATA EXPERIMENTS

Famous study of [Novembre et al. ‘08]: Take top two singular vectors of people x SNP matrix (POPRES)
REAL DATA EXPERIMENTS

Famous study of [Novembre et al. ‘08]: Take top two singular vectors of people x SNP matrix (POPRES)

“Genes Mirror Geography in Europe”
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?
Can we find such patterns in the presence of noise?

What PCA finds
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

What PCA finds
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise
RANSAC Projection

What RANSAC finds
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

XCS Projection

What robust PCA (via SDPs) finds
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise
Filter Projection

What our methods find
REAL DATA EXPERIMENTS

The power of provably robust estimation:

10% noise
Filter Projection

What our methods find
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

• Parameter Distance
• Detecting When an Estimator is Compromised
• A Win-Win Algorithm
• Unknown Covariance

Part III: Experiments

Part IV: The Levee Breaks
LIMITATIONS TO ROBUST ESTIMATION

Theorem [Diakonikolas, Kane, Stewart ‘16]: Any statistical query learning* algorithm that works against insertions and deletions that makes error $o(\epsilon \sqrt{\log 1/\epsilon})$ must make at least $d^{\omega(1)}$ queries
LIMITATIONS TO ROBUST ESTIMATION

Theorem [Diakonikolas, Kane, Stewart ‘16]: Any statistical query learning* algorithm in the strong corruption model that makes error $o(\epsilon \sqrt{\log 1/\epsilon})$ must make at least $d^{\omega(1)}$ queries

* Instead of seeing samples directly, an algorithm queries a function $f : \mathbb{R}^d \rightarrow [0, 1]$ and gets expectation, up to sampling noise
HANDLING MORE CORRUPTIONS

What if an adversary can corrupt the majority of samples?
HANDLING MORE CORRUPTIONS

What if an adversary can corrupt the majority of samples?

**Theorem [Charikar, Steinhardt, Valiant ‘17]:** Given samples from a distribution with mean $\mu$ and covariance $\Sigma \preceq \sigma^2 I$ where $1 - \alpha$ have been corrupted, there is an algorithm that outputs

$$\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_L$$

with $L \leq O\left(\frac{1}{\alpha}\right)$ that satisfies $\min_i \|\mu - \hat{\mu}_i\|_2 \leq O\left(\frac{\sigma}{\alpha^{1/2}}\right)$

This extends to mixtures straightforwardly
HANDLING MORE CORRUPTIONS

What if an adversary can corrupt the majority of samples?

**Theorem [Charikar, Steinhardt, Valiant ‘17]:** Given samples from a distribution with mean $\mu$ and covariance $\Sigma \preceq \sigma^2 I$ where $1 - \alpha$ have been corrupted, there is an algorithm that outputs

$$\hat{\mu}_1, \hat{\mu}_2, \ldots, \hat{\mu}_L$$

with $L \leq O\left(\frac{1}{\alpha}\right)$ that satisfies $\min_i \|\mu - \hat{\mu}_i\|_2 \leq O\left(\frac{\sigma}{\alpha^{1/2}}\right)$

This extends to mixtures straightforwardly

Guarantees were improved to $\min_i \|\mu - \hat{\mu}_i\|_2 \leq O\left(\frac{\sigma}{\alpha^{1/2}t}\right)$

in $d^{O(t)}$ time by [Diakonikolas et al ’18], [Kothari, Steinhardt ‘18]
Can we improve the sample complexity with sparsity assumptions?
SPARSE ROBUST ESTIMATION

Can we improve the sample complexity with sparsity assumptions?

Theorem [Li ‘17] [Du, Balakrishnan, Singh ’17]: There is an algorithm, in the unknown k-sparse mean case achieves error

$$\| \mu - \hat{\mu} \|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon})$$

with $N = O(k^2 \log d/\epsilon^2)$ samples.
SPARSE ROBUST ESTIMATION

Can we improve the sample complexity with sparsity assumptions?

Theorem [Li ‘17] [Du, Balakrishnan, Singh ’17]: There is an algorithm, in the unknown k-sparse mean case achieves error

$$\| \mu - \hat{\mu} \|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon})$$

with $N = O(k^2 \log d / \epsilon^2)$ samples

[Li ‘17] also studied robust sparse PCA
BACK TO THE CLASSICS

Applications to other classic learning problems?
BACK TO THE CLASSICS

Applications to other classic learning problems?

[Kothari, Steinardt ‘18] [Hopkins, Li ‘18] [Diakonikolas et al ‘18]:
There is an algorithm for learning spherical GMMs

\[ w_1 \mathcal{N}(\mu_1, \sigma^2 I) + \cdots + w_k \mathcal{N}(\mu_k, \sigma^2 I) \]

with separation \( k^\epsilon \) in time \( d^{O(1/\epsilon)} \)
BACK TO THE CLASSICS

Applications to other classic learning problems?

[Kothari, Steinardt ‘18] [Hopkins, Li ‘18] [Diakonikolas et al ‘18]: There is an algorithm for learning spherical GMMs

\[ w_1 \mathcal{N}(\mu_1, \sigma^2 I) + \cdots + w_k \mathcal{N}(\mu_k, \sigma^2 I) \]

with separation \( \kappa^\epsilon \) in time \( d^O(1/\epsilon) \)

[Klivans, Kothari, Meka ’18], [Diakonikolas et al ’18], [Du et al ’18], [Liu, Shen, Li, Caramanis ’18] gave algorithms for robust regression under various assumptions on distribution and loss function
LOOKING FORWARD

Can algorithms for agnostically learning a Gaussian help in exploratory data analysis in high-dimensions?

Isn’t this what we would have been doing with robust statistical estimators, if we had them all along?
LOOKING FORWARD

Can algorithms for agnostically learning a Gaussian help in exploratory data analysis in high-dimensions?

Isn’t this what we would have been doing with robust statistical estimators, if we had them all along?

What other fundamental tasks in high-dimensional statistics can be solved provably and robustly?
Summary:

• Dimension independent error bounds for robustly learning a Gaussian
• General recipe using restricted eigenvalue problems
• SQL lower bounds, handling more corruptions and sparse robust estimation

• Is practical, robust statistics within reach?
Summary:

- Dimension independent error bounds for robustly learning a Gaussian
- General recipe using restricted eigenvalue problems
- SQL lower bounds, handling more corruptions and sparse robust estimation

Is practical, robust statistics within reach?

Thanks! Any Questions?