Robust Estimators in High Dimensions, without the Computational Intractability

Ilias Diakonikolas\textsuperscript{1} Gautam Kamath\textsuperscript{2} Daniel Kane\textsuperscript{3}
Jerry Li\textsuperscript{2} Ankur Moitra\textsuperscript{2} Alistair Stewart\textsuperscript{1}

\textsuperscript{1}USC \textsuperscript{2}MIT \textsuperscript{3}USCD

June 1, 2016
Outline

- Introduction
- Detecting anomalous data
- The full algorithm
Outline

- **Introduction**
- Detecting anomalous data
- The full algorithm
Main Question

Is computationally efficient robust estimation possible in high dimensions?
Main Question

Is computationally efficient robust estimation possible in high dimensions?

Surprisingly poorly understood, even in very basic settings.
Main Question

Is computationally efficient robust estimation possible in high dimensions?

Surprisingly poorly understood, even in very basic settings.

This talk: A framework for computationally efficient robust estimation in high dimensions for a variety of fundamental learning tasks.
Noiseless case, 1-D

Given $n$ independent samples $X_1, \ldots, X_n$ from some Gaussian $\mathcal{N}(\mu, \sigma^2)$, how do we recover it?

Use empirical moments!

$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$

$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})^2$

Fact (folklore)

Given enough samples, these estimators converge provably to the truth with high probability, and are optimal.
Given $n$ independent samples $X_1, \ldots, X_n$ from some Gaussian $\mathcal{N}(\mu, \sigma^2)$, how do we recover it?

Use empirical moments!

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i$$
$$\hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})^2$$
Noiseless case, 1-D

Given $n$ independent samples $X_1, \ldots, X_n$ from some Gaussian $\mathcal{N}(\mu, \sigma^2)$, how do we recover it?

Use empirical moments!

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \hat{\sigma}^2 = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})^2$$

Fact (folklore)

Given enough samples, these estimators converge provably to the truth with high probability, and are optimal.
Noisy case, 1-D

Model misspecification: In the real world, things are often close to Gaussian, but (usually) never truly Gaussian.
Noisy case, 1-D

**Model misspecification:** In the real world, things are often close to Gaussian, but (usually) never truly Gaussian.
Noisy case, 1-D

**Model misspecification:** In the real world, things are often close to Gaussian, but (usually) never truly Gaussian.
Noisy case, 1-D

Model misspecification: In the real world, things are often close to Gaussian, but (usually) never truly Gaussian.
Noisy case, 1-D

**Model misspecification:** In the real world, things are often close to Gaussian, but (usually) never truly Gaussian.

**Question:** Are there estimators which work when you get samples from something which is not quite a Gaussian?
Huber’s contamination model

In this talk, we will consider a somewhat stronger notion of noise.
In this talk, we will consider a somewhat stronger notion of noise.

We get points from the true distribution, but an adversary gets to change an $\epsilon$-fraction of the samples *adversarially* before we see them.
In this talk, we will consider a somewhat stronger notion of noise.

We get points from the true distribution, but an adversary gets to change an \( \epsilon \)-fraction of the samples \textit{adversarially} before we see them.

**Definition (Huber’s contamination model)**

We say a set of samples is an \( \epsilon \)-corrupted set of samples of size \( n \) from a distribution \( F \) if they are generated via the following process:
In this talk, we will consider a somewhat stronger notion of noise.

We get points from the true distribution, but an adversary gets to change an $\epsilon$-fraction of the samples adversarially before we see them.

**Definition (Huber’s contamination model)**

We say a set of samples is an $\epsilon$-corrupted set of samples of size $n$ from a distribution $F$ if they are generated via the following process:

1. First generate $n$ independent samples from $F$. 


Huber’s contamination model

In this talk, we will consider a somewhat stronger notion of noise.

We get points from the true distribution, but an adversary gets to change an \( \epsilon \)-fraction of the samples \textit{adversarially} before we see them.

Definition (Huber’s contamination model)

We say a set of samples is an \( \epsilon \)-corrupted set of samples of size \( n \) from a distribution \( F \) if they are generated via the following process:

1. First generate \( n \) independent samples from \( F \).
2. An adversary then inspects these samples, and then changes an \( \epsilon \)-fraction of them arbitrarily.
Huber’s contamination model

In this talk, we will consider a somewhat stronger notion of noise.

We get points from the true distribution, but an adversary gets to change an $\epsilon$-fraction of the samples adversarially before we see them.

**Definition (Huber’s contamination model)**

We say a set of samples is an $\epsilon$-corrupted set of samples of size $n$ from a distribution $F$ if they are generated via the following process:

1. First generate $n$ independent samples from $F$.
2. An adversary then inspects these samples, and then changes an $\epsilon$-fraction of them arbitrarily.

**Goal:** Given an $\epsilon$-corrupted set of samples, recover $F$ as well as possible.
Huber’s contamination model
Huber’s contamination model
Huber’s contamination model
We will measure distance using total variation (TV) distance.
Formal Problem Statement

We will measure distance using total variation (TV) distance.

\[ d_{TV}(F, G) = \frac{1}{2} \int |f(x) - g(x)| dx. \]
Formal Problem Statement

We will measure distance using total variation (TV) distance.

\[ d_{TV}(F, G) = \frac{1}{2} \int |f(x) - g(x)| dx. \]

This will roughly correspond to parameter recovery with the appropriate scaling.
Formal Problem Statement

We will measure distance using total variation (TV) distance.

\[ d_{TV}(F, G) = \frac{1}{2} \int |f(x) - g(x)| dx. \]

This will roughly correspond to parameter recovery with the appropriate scaling.

Agnostic Learning

Let \( \mathcal{F} \) be a structured class of distributions (i.e. Gaussians, product distributions, etc.). Given an \( \epsilon \)-corrupted set of samples from some \( F \in \mathcal{F} \), output \( \hat{F} \) minimizing \( d_{TV}(F, \hat{F}) \).
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, \sigma^2)$. Can you learn $N$?
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?

Just need to learn the mean...
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?

Just need to learn the mean...

Q: Does the empirical mean still work?
Problem statement

Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?

Just need to learn the mean...

Q: Does the empirical mean still work? **No!**
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?

Just need to learn the mean...

Q: Does the empirical mean still work? **No!**
Learning a Gaussian, robustly

Problem statement
Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $N = \mathcal{N}(\mu, 1)$. Can you learn $N$?

Just need to learn the mean...

**Q:** Does the empirical mean still work? **No!**

But the empirical *median* does...
Fact (folklore)

Fix $\epsilon > 0$. Then given an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, 1)$ of sufficiently large size, then with probability at least 0.9,

$$|\mu - \text{median}| \leq O(\epsilon),$$

which implies

$$d_{TV}(\mathcal{N}(\mu, 1), \mathcal{N}(\text{median}, 1)) \leq O(\epsilon).$$

Moreover, this error is optimal up to constant factors.
Robust Statistics (cont.)

Fact (folklore)

Fix $\epsilon > 0$. Then given an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, 1)$ of sufficiently large size, then with probability at least 0.9,

$$|\mu - \text{median}| \leq O(\epsilon),$$

which implies

$$d_{TV}(\mathcal{N}(\mu, 1), \mathcal{N}(\text{median}, 1)) \leq O(\epsilon).$$

Moreover, this error is optimal up to constant factors.

Interquartile range (IQR) similarly works for recovering $\sigma^2$. 
Robust Statistics (cont.)
Robust Statistics (cont.)

Median and IQR are robust statistics
Robust Statistics (cont.)

Median and IQR are **robust statistics**

Classical subfield of statistics [HR09]
Robust Statistics (cont.)

Median and IQR are robust statistics

Classical subfield of statistics [HR09]
Robust Statistics

Three things to worry about:
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
   - Ideally should be $\Theta(\epsilon)$.
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
   - Ideally should be $\Theta(\epsilon)$.

2. **Runtime:** How fast can I compute my robust estimator?
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
   - Ideally should be $\Theta(\epsilon)$.

2. **Runtime:** How fast can I compute my robust estimator?

3. **Sample complexity:** How many samples do I need to get this guarantee?
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
   - Ideally should be $\Theta(\epsilon)$.

2. **Runtime:** How fast can I compute my robust estimator?

3. **Sample complexity:** How many samples do I need to get this guarantee?

These issues are well-studied and generally well-understood.
Robust Statistics

Three things to worry about:

1. **Recovery guarantee:** Given an $\epsilon$-corrupted set of samples of sufficiently large size, what is the error I achieve?
   - Ideally should be $\Theta(\epsilon)$.

2. **Runtime:** How fast can I compute my robust estimator?

3. **Sample complexity:** How many samples do I need to get this guarantee?

These issues are well-studied and generally well-understood...in low dimensions
High Dimensional Robust Statistics?

Nowadays, many natural ML applications involve extremely high dimensional data.
High Dimensional Robust Statistics?

Nowadays, many natural ML applications involve extremely high dimensional data.

In this setting, robust statistics are less well understood.
Nowadays, many natural ML applications involve extremely high dimensional data.

In this setting, robust statistics are less well understood.

**Problem Statement:**
Given an $\epsilon$-corrupted set of samples from some $d$-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$, recover $\hat{\mu}, \hat{\Sigma}$ minimizing $d_{TV}(\mathcal{N}(\mu, \Sigma), \mathcal{N}(\hat{\mu}, \hat{\Sigma}))$. 
Nowadays, many natural ML applications involve extremely high dimensional data.

In this setting, robust statistics are less well understood.

**Problem Statement:**

Given an $\epsilon$-corrupted set of samples from some $d$-dimensional Gaussian $\mathcal{N}(\mu, I)$, recover $\hat{\mu}$ minimizing $d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I))$. 
High Dimensional Robust Statistics?

Nowadays, many natural ML applications involve extremely high dimensional data.

In this setting, robust statistics are less well understood.

**Problem Statement:**

Given an $\epsilon$-corrupted set of samples from some $d$-dimensional Gaussian $\mathcal{N}(\mu, I)$, recover $\hat{\mu}$ minimizing $d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I))$.

Equivalently, find $\hat{\mu}$ minimizing $\|\mu - \hat{\mu}\|_2$. 
## High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>(O(\epsilon))</td>
<td>(NP-hard)</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>(O(\epsilon \sqrt{d}))</td>
<td>(poly time)</td>
</tr>
<tr>
<td>Tournament</td>
<td>(O(\epsilon n))</td>
<td>(O(d))</td>
</tr>
<tr>
<td>Pruning</td>
<td>(O(\epsilon \sqrt{d}))</td>
<td>(O(n))</td>
</tr>
<tr>
<td>RANSAC</td>
<td>(O(\infty))</td>
<td>(O(n))</td>
</tr>
<tr>
<td>etc...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon \sqrt{d})$</td>
<td>$O(n^{O(d)})$</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>---------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$</td>
<td>$O(n)$</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$</td>
<td></td>
</tr>
</tbody>
</table>

The Tukey Median algorithm has a runtime of $O(n)$ with an error guarantee of $O(\epsilon)$. Other algorithms like Geometric Median, Tournament Pruning, RANSAC, etc. have different complexities.
## High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$</td>
<td>NP-hard</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>( O(\epsilon) ) ✓</td>
<td>NP-hard ❌</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>-----------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\varepsilon)$</td>
<td>NP-hard</td>
</tr>
</tbody>
</table>

Geometric Median
<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✓</td>
<td>NP-hard ✗</td>
</tr>
<tr>
<td>Geometric Median</td>
<td></td>
<td>poly time</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</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>poly time ✓</td>
</tr>
</tbody>
</table>
## High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✓</td>
<td>NP-hard ✗</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$</td>
<td>poly time ✓</td>
</tr>
</tbody>
</table>
### High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$</td>
<td>NP-hard</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$</td>
<td>poly time</td>
</tr>
<tr>
<td>Algorithm</td>
<td>Error Guarantee</td>
<td>Runtime</td>
</tr>
<tr>
<td>--------------------</td>
<td>-----------------</td>
<td>--------------</td>
</tr>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✔️</td>
<td>NP-hard ❌</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$ ❌</td>
<td>poly time ✔️</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\epsilon)$ ✔️</td>
<td>$n^{O(d)}$ ❌</td>
</tr>
</tbody>
</table>
## High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✔</td>
<td>NP-hard ✗</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>poly time ✔</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\epsilon)$ ✔</td>
<td>$n^{O(d)}$ ✗</td>
</tr>
<tr>
<td>Pruning</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>$O(n)$ ✔</td>
</tr>
</tbody>
</table>
### High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✓</td>
<td>NP-hard ✗</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>poly time ✓</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\epsilon)$ ✓</td>
<td>$n^{O(d)}$ ✗</td>
</tr>
<tr>
<td>Pruning</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>$O(n)$ ✓</td>
</tr>
<tr>
<td>RANSAC</td>
<td>$O(\infty)$ ? ✗</td>
<td>$O(n)$ ✓</td>
</tr>
</tbody>
</table>
## High Dimensional Robust Statistics? (cont.)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Error Guarantee</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tukey Median</td>
<td>$O(\epsilon)$ ✓</td>
<td>NP-hard ✗</td>
</tr>
<tr>
<td>Geometric Median</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>poly time ✓</td>
</tr>
<tr>
<td>Tournament</td>
<td>$O(\epsilon)$ ✓</td>
<td>$n^{O(d)}$ ✗</td>
</tr>
<tr>
<td>Pruning</td>
<td>$O(\epsilon \sqrt{d})$ ✗</td>
<td>$O(n)$ ✓</td>
</tr>
<tr>
<td>RANSAC</td>
<td>$O(\infty)$ ? ✗</td>
<td>$O(n)$ ✓</td>
</tr>
<tr>
<td>etc...</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
All known robust estimators for the mean in high dimensions are either:
- computationally intractable, or
- lose a dimension-dependent factor in the error.

Question

Can you agnostically learn a Gaussian in high dimensions efficiently, with
no dimension dependence in the error?
All known robust estimators for the mean in high dimensions are either:

- computationally intractable,
- lose a dimension-dependent factor in the error.
All known robust estimators for the mean in high dimensions are either:

- computationally intractable, or
All known robust estimators for the mean in high dimensions are either:

- computationally intractable, or

- lose a dimension-dependent factor in the error

Question: Can you agnostically learn a Gaussian in high dimensions efficiently, with no dimension dependence in the error?
All known robust estimators for the mean in high dimensions are either:

- computationally intractable, or
- lose a dimension-dependent factor in the error

**Question**

Can you agnostically learn a Gaussian in high dimensions efficiently, with no dimension dependence in the error?
Our Results

We give the first robust estimator for Gaussians in high dimensions that is:

- efficiently computable ✔
- loses no dimension factors in the error ✔

**Theorem**

*Fix $\epsilon > 0$. There is an algorithm, which given an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$ of sufficiently large size, outputs $\hat{\mu}, \hat{\Sigma}$ so that with probability at least 0.9,*

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

*Moreover, this error is optimal up to logarithmic factors.*
Our Results

We give the first robust estimator for Gaussians in high dimensions that is:

- efficiently computable ✓
- loses no dimension factors in the error ✓

Theorem

Fix $\epsilon > 0$. There is an algorithm, which given an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$ of sufficiently large size, outputs $\hat{\mu}, \hat{\Sigma}$ so that with probability at least 0.9,

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

Moreover, this error is optimal up to logarithmic factors.

We also give similar results for product distributions, mixtures of balanced product distributions, and mixtures of spherical Gaussians.
Our Results

We give the first robust estimator for Gaussians in high dimensions that is:

- efficiently computable ✓
- loses no dimension factors in the error ✓

Theorem

Fix $\epsilon > 0$. There is an algorithm, which given an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, \Sigma)$ of sufficiently large size, outputs $\hat{\mu}, \hat{\Sigma}$ so that with probability at least 0.9,

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

Moreover, this error is optimal up to logarithmic factors.

We also give similar results for product distributions, mixtures of balanced product distributions, and mixtures of spherical Gaussians.

Concurrent work of [LRV15] gives an algorithm for the same problem, but with somewhat worse parameters.
Outline

- Introduction
- Detecting anomalous data
- The full algorithm
Learning the mean, robustly

**Problem Statement:** Given an $\epsilon$-corrupted set of samples from some $d$-dimensional Gaussian $N(\mu, I)$, recover $\hat{\mu}$ minimizing $d_{\text{TV}}(N(\mu, I), N(\hat{\mu}, I))$. 
Learning the mean, robustly

**Problem Statement:** Given an $\epsilon$-corrupted set of samples from some $d$-dimensional Gaussian $\mathcal{N}(\mu, I)$, recover $\hat{\mu}$ minimizing $d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I))$.

**Fact (folklore)**

Let $\epsilon > 0$ be sufficiently small. Then $d_{TV}(\mathcal{N}(\mu, I), \mathcal{N}(\hat{\mu}, I)) < \epsilon$ if and only if $\|\mu - \hat{\mu}\|_2 \leq O(\epsilon)$. 
Why do naive methods get stuck?

Consider the following, simple algorithm:
Why do naive methods get stuck?

Consider the following, simple algorithm:

**Naive Pruning**

Remove all points which are obviously too far away to be from the Gaussian, then take the empirical mean of the remaining points.
Why do naive methods get stuck?

Consider the following, simple algorithm:

Naive Pruning
Remove all points which are obviously too far away to be from the Gaussian, then take the empirical mean of the remaining points.
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

Fact

Let \( X \sim \mathcal{N}(\mu, 1) \). Then

\[
\Pr(\mid X - \mu \mid > \sqrt{\log \frac{1}{\delta}}) < O(\delta).
\]

So if you truncate at \( O(\sqrt{\log \frac{1}{\epsilon}}) \), then:

You throw away at most an \( O(\epsilon) \) fraction of the good points.

All remaining bad points are at most \( O(\sqrt{\log \frac{1}{\epsilon}}) \) away from \( \mu \).

\( \Rightarrow \) The empirical mean of the truncated points satisfies

\[
\mid \hat{\mu} - \mu \mid < O(\epsilon \sqrt{\log \frac{1}{\epsilon}}).
\]
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

**Fact**

Let $X \sim \mathcal{N}(\mu, 1)$. Then $\Pr(|X - \mu| > \sqrt{\log 1/\delta}) < O(\delta)$.
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

**Fact**

Let $X \sim \mathcal{N}(\mu, 1)$. Then $\Pr(|X - \mu| > \sqrt{\log 1/\delta}) < O(\delta)$. 

So if you truncate at $O(\sqrt{\log 1/\epsilon})$, then:
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

Fact

Let \( X \sim \mathcal{N}(\mu, 1) \). Then \( \Pr(|X - \mu| > \sqrt{\log 1/\delta}) < O(\delta) \).

So if you truncate at \( O(\sqrt{\log 1/\epsilon}) \), then:

- You throw away at most an \( O(\epsilon) \) fraction of the good points
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

**Fact**

Let \( X \sim \mathcal{N}(\mu, 1) \). Then \( \Pr(|X - \mu| > \sqrt{\log 1/\delta}) < O(\delta) \).

So if you truncate at \( O(\sqrt{\log 1/\varepsilon}) \), then:

- You throw away at most an \( O(\varepsilon) \) fraction of the good points
- All remaining bad points are at most \( O(\sqrt{\log 1/\varepsilon}) \) away from \( \mu \).
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In 1 dimension:

Fact

\[
\text{Let } X \sim \mathcal{N}(\mu, 1). \text{ Then } \Pr(|X - \mu| > \sqrt{\log \frac{1}{\delta}}) < O(\delta).
\]

So if you truncate at \(O(\sqrt{\log \frac{1}{\epsilon}})\), then:

- You throw away at most an \(O(\epsilon)\) fraction of the good points
- All remaining bad points are at most \(O(\sqrt{\log \frac{1}{\epsilon}})\) away from \(\mu\).

\[\Rightarrow\] The empirical mean of the truncated points satisfies

\[
|\hat{\mu} - \mu| < O(\epsilon \sqrt{\log \frac{1}{\epsilon}}).
\]
Why do naive methods get stuck? (cont.)

What does “obviously too far away” mean?

In $d$ dimensions:

**Fact**

Let $X \sim \mathcal{N}(\mu, I)$. Then $\Pr(\|X - \mu\|_2 > \sqrt{d \log 1/\delta}) < O(\delta)$. 

So if you truncate at $O(\sqrt{d \log 1/\epsilon})$, then:

- You throw away at most an $O(\epsilon)$ fraction of the good points
- All remaining bad points are at most $O(\sqrt{d \log 1/\epsilon})$ away from $\mu.$

$\Rightarrow$ The empirical mean of the truncated points satisfies

$$\|\hat{\mu} - \mu\|_2 < O(\epsilon \sqrt{d \log 1/\epsilon}).$$
Why do naive methods get stuck? (cont.)
Why do naive methods get stuck? (cont.)
Detecting anomalous data

Why do naive methods get stuck? (cont.)

\[ \sqrt{d} \]
Why do naive methods get stuck? (cont.)

Any method that only looks for outliers individually will get stuck at $O(\epsilon \sqrt{d})$. 
Why do naive methods get stuck? (cont.)

Any method that only looks for outliers individually will get stuck at $O(\epsilon \sqrt{d})$.

We look for corruptions on a global scale.
Detecting anomalous data

Why do naive methods get stuck? (cont.)

Any method that only looks for outliers individually will get stuck at $O(\epsilon \sqrt{d})$.

We look for corruptions on a global scale.

**Our idea:** If the corrupted points move the empirical mean by more than $\tilde{O}(\epsilon)$, then they must also screw up the empirical covariance!
Main Proposition

**Proposition**

Fix $\epsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size $n$, where $n = \Omega \left( \frac{d + \log 1/\delta}{\epsilon^2} \right)$. 
Detecting anomalous data

Main Proposition

Proposition

Fix $\epsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size $n$, where $n = \Omega \left( \frac{d + \log 1/\delta}{\epsilon^2} \right)$. Let

\[
\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \text{and} \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T.
\]
Main Proposition

Proposition

Fix $\epsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $\mathcal{N}(\mu, I)$ of size $n$, where $n = \Omega \left( \frac{d + \log 1/\delta}{\epsilon^2} \right)$. Let

$$\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T.$$ 

With probability $1 - \delta$, if $\|\mu - \hat{\mu}\|_2 > \Omega(\epsilon \sqrt{\log 1/\epsilon})$, then

$$\|\hat{\Sigma} - I\|_2 \geq \Omega \left( \frac{\|\mu - \hat{\mu}\|^2_2}{\epsilon} \right) \geq \Omega(\epsilon \log 1/\epsilon).$$
Detecting anomalous data

Main Proposition

**Proposition**

Fix $\epsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $\mathcal{N} (\mu, \mathbf{I})$ of size $n$, where $n = \Omega \left( \frac{d + \log 1/\delta}{\epsilon^2} \right)$. Let

$$ \hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T. $$

With probability $1 - \delta$, if $\| \mu - \hat{\mu} \|_2 > \Omega (\epsilon \sqrt{\log 1/\epsilon})$, then

$$ \| \hat{\Sigma} - \mathbf{I} \|_2 \geq \Omega \left( \frac{\| \mu - \hat{\mu} \|^2_2}{\epsilon} \right) \geq \Omega (\epsilon \log 1/\epsilon) $$
Main Proposition

Proposition

Fix $\epsilon, \delta > 0$. Let $X_1, \ldots, X_n$ be an $\epsilon$-corrupted set of samples from $\mathcal{N}(0, I)$ of size $n$, where $n = \Omega \left( \frac{d + \log 1/\delta}{\epsilon^2} \right)$. Let

$$
\hat{\mu} = \frac{1}{n} \sum_{i=1}^{n} X_i \quad \hat{\Sigma} = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T.
$$

With probability $1 - \delta$, if $\|\hat{\mu}\|_2 > \Omega(\epsilon \sqrt{\log 1/\epsilon})$, then

$$
\|\hat{\Sigma} - I\|_2 \geq \Omega \left( \frac{\|\hat{\mu}\|_2^2}{\epsilon} \right) \geq \Omega(\epsilon \log 1/\epsilon)
$$
Proof of the Main Proposition: Preliminaries
Proof of the Main Proposition: Preliminaries

Let $S_{\text{bad}}$ be the indices of the corrupted points, and let $S_{\text{good}}$ be the remaining set of indices.
Proof of the Main Proposition: Preliminaries

Let $S_{\text{bad}}$ be the indices of the corrupted points, and let $S_{\text{good}}$ be the remaining set of indices. Recall $|S_{\text{bad}}| = \epsilon n$. 

Lemma
With probability $1 - O(\delta)$, we have

$$
\| \sum_{i \in S_{\text{good}}} x_i \|^2 \leq O(\epsilon \sqrt{\log 1/\epsilon}).
$$

Lemma
With probability $1 - O(\delta)$, we have

$$
\| \sum_{i \in S_{\text{good}}} x_i x_i^T - I \|^2 \leq O(\epsilon \sqrt{\log 1/\epsilon}).
$$
Proof of the Main Proposition: Preliminaries

Let $S_{\text{bad}}$ be the indices of the corrupted points, and let $S_{\text{good}}$ be the remaining set of indices. Recall $|S_{\text{bad}}| = \epsilon n$.

**Lemma**

*With probability* $1 - O(\delta)$, *we have*

$$
\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon}).
$$
Proof of the Main Proposition: Preliminaries

Let $S_{\text{bad}}$ be the indices of the corrupted points, and let $S_{\text{good}}$ be the remaining set of indices. Recall $|S_{\text{bad}}| = \epsilon n$. 

**Lemma**

*With probability $1 - O(\delta)$, we have*

$$\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon}).$$

**Lemma**

*With probability $1 - O(\delta)$, we have*

$$\left\| \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i X_i^T - I \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon}).$$
Proof of the Main Proposition (cont.)

For the rest of the proof, we will condition on the previous two inequalities both holding.
Proof of the Main Proposition (cont.)

For the rest of the proof, we will condition on the previous two inequalities both holding.

**Lemma**

If \( \| \hat{\mu} \|_2 \geq \Omega(\epsilon \sqrt{\log 1/\epsilon}) \), then

\[
\left\| \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i X_i^T \right\|_2 \geq \Omega \left( \frac{\| \hat{\mu} \|_2^2}{\epsilon^2} \right).
\]
Proof of Lemma

By assumption, we have
Proof of Lemma

By assumption, we have

$$\|\hat{\mu}\|_2$$
Proof of Lemma

By assumption, we have

$$
\|\hat{\mu}\|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2
$$
Proof of Lemma

By assumption, we have

\[ \| \hat{\mu} \|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \geq \Omega(\epsilon \sqrt{\log 1/\epsilon}) \]
Proof of Lemma

By assumption, we have

\[
\| \hat{\mu} \|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \geq \Omega(\epsilon \sqrt{\log 1/\epsilon})
\]

\[
\leq O(\epsilon \sqrt{\log 1/\epsilon})
\]
Proof of Lemma

By assumption, we have

\[ \|\hat{\mu}\|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon}) \]

\[ \Rightarrow \left\| \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \geq \Omega(\|\hat{\mu}\|_2) \]
Proof of Lemma

By assumption, we have

\[ \| \hat{\mu} \|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon}) \]

\[ \Rightarrow \left\| S_{\text{bad}} \right\| \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \geq \Omega(\| \hat{\mu} \|_2) \]
By assumption, we have

\[
\| \hat{\mu} \|_2 = \left\| \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \leq O(\epsilon \sqrt{\log 1/\epsilon})
\]

\[
\Rightarrow \left\| \frac{|S_{\text{bad}}|}{n} \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \geq \Omega(\| \hat{\mu} \|_2)
\]

\[
\Rightarrow \left\| \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i \right\|_2 \geq \Omega \left( \frac{\| \hat{\mu} \|_2}{\epsilon} \right)
\]
Proof of Lemma (cont.)

Fact (PSDness of covariance)

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be arbitrary. Then

$$\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.$$
Fact (PSDness of covariance)

Let \( \mathbf{v}_1, \ldots, \mathbf{v}_m \) be arbitrary. Then

\[
\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.
\]

Natural generalization of the fact that \( \mathbb{E}[X^2] \geq \mathbb{E}[X]^2 \).
Proof of Lemma (cont.)

Fact (PSDness of covariance)

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be arbitrary. Then

$$
\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.
$$

Natural generalization of the fact that $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$.

$$
\Rightarrow \quad \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i X_i^T \succeq \left( \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i \right) \left( \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i \right)^T.
$$
Proof of Lemma (cont.)

Fact (PSDness of covariance)

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be arbitrary. Then

$$\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.$$ 

Natural generalization of the fact that $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$.

$$\Rightarrow \left\| \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i X_i^T \right\|_2 \geq \left\| \left( \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i \right) \left( \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i \right)^T \right\|_2.$$
Proof of Lemma (cont.)

Fact (PSDness of covariance)

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be arbitrary. Then

$$
\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.
$$

Natural generalization of the fact that $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$.

$$
\Rightarrow \quad \left\| \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i X_i^T \right\|_2 \geq \left\| \left( \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i \right) \right\|_2^2
$$
Proof of Lemma (cont.)

Fact (PSDness of covariance)

Let $\mathbf{v}_1, \ldots, \mathbf{v}_m$ be arbitrary. Then

$$
\frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \mathbf{v}_i^T \succeq \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right) \left( \frac{1}{m} \sum_{i=1}^{m} \mathbf{v}_i \right)^T.
$$

Natural generalization of the fact that $\mathbb{E}[X^2] \geq \mathbb{E}[X]^2$.

$$
\Rightarrow \left\| \frac{1}{|\mathcal{S}_{\text{bad}}|} \sum_{i \in \mathcal{S}_{\text{bad}}} X_i X_i^T \right\|_2 \geq \left\| \left( \frac{1}{|\mathcal{S}_{\text{bad}}|} \sum_{i \in \mathcal{S}_{\text{bad}}} X_i \right) \right\|_2^2 \geq \Omega \left( \frac{\|\hat{\mu}\|_2^2}{\epsilon^2} \right). \quad \square
$$
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n}(X_i - \hat{\mu})(X_i - \hat{\mu})^T - I \\
= \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{1}{n} \sum_{i \in S_{\text{good}}} X_iX_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i X_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{|S_{\text{good}}|}{n} \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i X_i^T + \frac{|S_{\text{bad}}|}{n} \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]
Proof of Main Proposition (cont.)

Recall:

\[ \hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I \]

\[ = \frac{1}{n} \sum_{i=1}^{n} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = \frac{1}{n} \sum_{i \in S_{\text{good}}} X_iX_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = (1 - \epsilon) \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_iX_i^T + \epsilon \cdot \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I \]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{1}{n} \sum_{i \in S_{good}} X_iX_i^T + \frac{1}{n} \sum_{i \in S_{bad}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= (1 - \epsilon) \left( \frac{1}{|S_{good}|} \sum_{i \in S_{good}} X_iX_i^T - I \right) - \epsilon I + \epsilon \cdot \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_iX_i^T - \hat{\mu}\hat{\mu}^T
\]
Proof of Main Proposition (cont.)

Recall:

\[ \hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I \]

\[ = \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i X_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = (1 - \epsilon) \left( \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_i X_i^T - I \right) + \epsilon I + \epsilon \cdot \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T \]

\[ \geq \frac{\|\hat{\mu}\|_2^2}{\epsilon} \geq \Omega\left(\epsilon \log \frac{1}{\epsilon}\right) \]
Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{1}{n} \sum_{i \in S_{good}} X_i X_i^T + \frac{1}{n} \sum_{i \in S_{bad}} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= (1 - \epsilon) \left( \frac{1}{|S_{good}|} \sum_{i \in S_{good}} X_i X_i^T - I \right) + \epsilon I + \epsilon \cdot \frac{1}{|S_{bad}|} \sum_{i \in S_{bad}} X_i X_i^T - \hat{\mu}\hat{\mu}^T
\]

\[
\leq O(\epsilon \sqrt{\log 1/\epsilon})
\]

\[
\geq \frac{\|\hat{\mu}\|^2}{\epsilon} \geq \Omega(\epsilon \log 1/\epsilon)
\]
Proof of Main Proposition (cont.)

Recall:

\[ \hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I \]

\[ = \frac{1}{n} \sum_{i=1}^{n} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = \frac{1}{n} \sum_{i \in S_{\text{good}}} X_iX_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I \]

\[ = (1 - \epsilon) \left( \frac{1}{|S_{\text{good}}|} \sum_{i \in S_{\text{good}}} X_iX_i^T - I \right) + \epsilon I \leq O(\epsilon) \]

\[ + \epsilon \cdot \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T \geq \frac{\|\hat{\mu}\|^2}{\epsilon} \geq \Omega(\epsilon \log 1/\epsilon) \]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

\[
= \frac{1}{n} \sum_{i=1}^{n} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= \frac{1}{n} \sum_{i \in S_{\text{good}}} X_iX_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T - I
\]

\[
= (1 - \epsilon) \frac{1}{|S_{\text{good}}|} \left( \sum_{i \in S_{\text{good}}} X_iX_i^T - I \right) + \epsilon I \leq O(\epsilon) + \epsilon \cdot \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_iX_i^T - \hat{\mu}\hat{\mu}^T \leq \|\hat{\mu}\|_2^2
\]

\[
\leq O(\epsilon \sqrt{\log 1/\epsilon}) \quad \geq \frac{\|\hat{\mu}\|_2^2}{\epsilon} \geq \Omega(\epsilon \log 1/\epsilon)
\]
Proof of Main Proposition (cont.)

Recall:

\[
\hat{\Sigma} - I = \frac{1}{n} \sum_{i=1}^{n} (X_i - \hat{\mu})(X_i - \hat{\mu})^T - I
\]

= \frac{1}{n} \sum_{i=1}^{n} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I

= \frac{1}{n} \sum_{i \in S_{\text{good}}} X_i X_i^T + \frac{1}{n} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T - I

= (1 - \epsilon) \frac{1}{|S_{\text{good}}|} \left( \sum_{i \in S_{\text{good}}} X_i X_i^T - I \right) + \epsilon I + \epsilon \cdot \frac{1}{|S_{\text{bad}}|} \sum_{i \in S_{\text{bad}}} X_i X_i^T - \hat{\mu}\hat{\mu}^T

\leq O(\epsilon \sqrt{\log 1/\epsilon}) + \epsilon I

\geq \frac{\|\hat{\mu}\|_2^2}{\epsilon} \geq \Omega(\epsilon \log 1/\epsilon)

\Rightarrow \|\hat{\Sigma} - I\|_2 \geq \Omega \left( \frac{\|\mu\|_2^2}{\epsilon} \right), \text{ as claimed.} \quad \square
Summary so far

We can detect whether or not the empirical mean is off from what it ought to be by checking the spectrum of the empirical covariance!
We can detect whether or not the empirical mean is off from what it ought to be by checking the spectrum of the empirical covariance!

But how do we use this algorithmically?
Outline

- Introduction
- Detecting anomalous data
- The full algorithm
Our Approaches

We give two algorithms motivated by the intuition given previously:

1. **Filtering**: Runs in $\tilde{O}(n^{1/2})$ time, but requires more samples.
2. **Convex programming**: Runs in $\text{poly}(d, n, 1/\epsilon)$ time, but is sample optimal.

**Shared idea:** If $\|\hat{\Sigma} - I\|_2$ is large, then the top eigenvector of the empirical covariance gives a direction in which the corrupted points dominate!
Our Approaches

We give two algorithms motivated by the intuition given previously:

1. **Filtering:** Runs in $\tilde{O}(nd^2)$ time, but requires more samples
Our Approaches

We give two algorithms motivated by the intuition given previously:

1. **Filtering**: Runs in $\tilde{O}(nd^2)$ time, but requires more samples

2. **Convex programming**: Runs in $\text{poly}(d, n, 1/\epsilon)$ time, but is sample optimal
Our Approaches

We give two algorithms motivated by the intuition given previously:

1. **Filtering:** Runs in $\tilde{O}(nd^2)$ time, but requires more samples

2. **Convex programming:** Runs in $\text{poly}(d, n, 1/\epsilon)$ time, but is sample optimal

**Shared idea:** If $\|\hat{\Sigma} - I\|_2$ is large, then the top eigenvector of the empirical covariance gives a direction in which the corrupted points dominate!
Approach 1: Filtering

While $\|\hat{\Sigma} - I\|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:

Let $v$ be (approximately) the top eigenvector of $\hat{\Sigma}$.

One can show that we can find some $T$ so that

$|\{i \in S_{\text{bad}} : |X_i \cdot v| > T\}| > |\{i \in S_{\text{good}} : |X_i \cdot v| > T\}|$.

Throw out all points with $|X_i \cdot v| > T$.

Return $\hat{\mu}$, the empirical mean of the remaining points.
Approach 1: Filtering

While $\| \hat{\Sigma} - I \|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:

- Let $v$ be (approximately) the top eigenvector of $\hat{\Sigma}$.
- One can show that we can find some $T$ so that $|\{ i \in S_{\text{bad}} : |X_i \cdot v| > T \}| > |\{ i \in S_{\text{good}} : |X_i \cdot v| > T \}|$.
- Throw out all points with $|X_i \cdot v| > T$.
- Return $\hat{\mu}$, the empirical mean of the remaining points.
Approach 1: Filtering

While $\|\Sigma - I\|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:
- Let $v$ be (approximately) the top eigenvector of $\Sigma$. 

One can show that we can find some $T$ so that $|\{i \in S_{bad} : |X_i \cdot v| > T\}| > |\{i \in S_{good} : |X_i \cdot v| > T\}|$.

Throw out all points with $|X_i \cdot v| > T$.

Return $\hat{\mu}$, the empirical mean of the remaining points.
Approach 1: Filtering

While $\|\hat{\Sigma} - I\|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:

- Let $\mathbf{v}$ be (approximately) the top eigenvector of $\hat{\Sigma}$.

- One can show that we can find some $T$ so that

$$\left| \{i \in S_{\text{bad}} : |X_i \cdot \mathbf{v}| > T \} \right| > \left| \{i \in S_{\text{good}} : |X_i \cdot \mathbf{v}| > T \} \right|.$$

- Throw out all points with $|X_i \cdot \mathbf{v}| > T$.

- Return $\hat{\mu}$, the empirical mean of the remaining points.
Approach 1: Filtering

While $\|\hat{\Sigma} - I\|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:

- Let $\mathbf{v}$ be (approximately) the top eigenvector of $\hat{\Sigma}$.

- One can show that we can find some $T$ so that

$$\left| \left\{ i \in S_{\text{bad}} : |X_i \cdot \mathbf{v}| > T \right\} \right| > \left| \left\{ i \in S_{\text{good}} : |X_i \cdot \mathbf{v}| > T \right\} \right| .$$

- Throw out all points with $|X_i \cdot \mathbf{v}| > T$. 

Return $\hat{\mu}$, the empirical mean of the remaining points.
Approach 1: Filtering

While $\|\hat{\Sigma} - I\|_2 \geq \Omega(\epsilon \log 1/\epsilon)$:

- Let $\mathbf{v}$ be (approximately) the top eigenvector of $\hat{\Sigma}$.

- One can show that we can find some $T$ so that

\[
|\{i \in S_{\text{bad}} : |X_i \cdot \mathbf{v}| > T\}| > |\{i \in S_{\text{good}} : |X_i \cdot \mathbf{v}| > T\}|.
\]

- Throw out all points with $|X_i \cdot \mathbf{v}| > T$.

- Return $\hat{\mu}$, the empirical mean of the remaining points.
The full algorithm

Approach 1: Filtering (cont.)

Algorithm runs in $\tilde{O}(d^2)$ time—potentially practical! One can show that the loop only runs for $O(d)$ iterations, and each iteration can be done in $\tilde{O}(nd)$ time.

Algorithm is a bit aggressive in throwing away samples—seems to require $O(d^2/\epsilon^2)$ samples.

Requires concentration of LTFs in Gaussian space. We conjecture this can be improved to $O(d^2/\epsilon^2 + d^2/\epsilon^2)$. 
Approach 1: Filtering (cont.)

- The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!
The full algorithm

Approach 1: Filtering (cont.)

- The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!
The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!

One can show that the loop only runs for $O(d)$ iterations, and each iteration can be done in $\tilde{O}(nd)$ time.
Approach 1: Filtering (cont.)

- The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!
  - One can show that the loop only runs for $O(d)$ iterations, and each iteration can be done in $\tilde{O}(nd)$ time.

- Algorithm is a bit aggressive in throwing away samples—seems to require $O(d^2/\epsilon^2)$ samples.
Approach 1: Filtering (cont.)

- The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!
  - One can show that the loop only runs for $O(d)$ iterations, and each iteration can be done in $\tilde{O}(nd)$ time.

- Algorithm is a bit aggressive in throwing away samples—seems to require $O(d^2/\epsilon^2)$ samples.
  - Requires concentration of LTFs in Gaussian space
The full algorithm

Approach 1: Filtering (cont.)

- The algorithm runs in $\tilde{O}(nd^2)$ time—potentially practical!
  - One can show that the loop only runs for $O(d)$ iterations, and each iteration can be done in $\tilde{O}(nd)$ time.

- Algorithm is a bit aggressive in throwing away samples—seems to require $O(d^2/\epsilon^2)$ samples.
  - Requires concentration of LTFs in Gaussian space
  - We conjecture this can be improved to $O(d/\epsilon^2 + d^2/\epsilon)$. 
Approach 2: Convex programming
Approach 2: Convex programming

**Idea:** Give weights to each sample point, and adjust the weights if we think they’re contributing to the noise.
Approach 2: Convex programming

**Idea:** Give weights to each sample point, and adjust the weights if we think they’re contributing to the noise.

For any $I \subseteq [n]$, let $w^I$ be so that $(w^I)_j = 1/|I|$ if $j \in I$ and 0 otherwise.
Approach 2: Convex programming

**Idea:** Give weights to each sample point, and adjust the weights if we think they’re contributing to the noise.

For any $I \subseteq [n]$, let $w^I$ be so that $(w^I)_j = 1/|I|$ if $j \in I$ and 0 otherwise. Let

$$ S_{n, \epsilon} = \text{conv} \left\{ w^I : |I| = (1 - \epsilon)n \right\}, $$
Approach 2: Convex programming

Idea: Give weights to each sample point, and adjust the weights if we think they’re contributing to the noise.

For any $I \subseteq [n]$, let $w^I$ be so that $(w^I)_j = 1/|I|$ if $j \in I$ and 0 otherwise. Let

$$S_{n, \epsilon} = \text{conv} \left\{ w^I : |I| = (1 - \epsilon)n \right\},$$

$$C_\delta = \left\{ w \in S_{n, \epsilon} : \left\| \sum_{i=1}^n w_i (X_i - \mu)(X_i - \mu)^T - I \right\|_2 \leq \delta \right\}. $$
Approach 2: Convex programming

Idea: Give weights to each sample point, and adjust the weights if we think they’re contributing to the noise.

For any \( I \subseteq [n] \), let \( w^I \) be so that \((w^I)_j = 1/|I|\) if \( j \in I \) and 0 otherwise. Let

\[
S_{n,\epsilon} = \text{conv} \left\{ w^I : |I| = (1 - \epsilon)n \right\},
\]

\[
C_\delta = \left\{ w \in S_{n,\epsilon} : \left\| \sum_{i=1}^{n} w_i(X_i - \mu)(X_i - \mu)^T - I \right\|_2 \leq \delta \right\}.
\]

\( C_\delta \) is an unknown convex set, however, we show we can give a separation oracle for it \( \Rightarrow \) we can find a point in this set via the ellipsoid algorithm.
Approach 2: Convex programming (cont.)

The algorithm is very careful with samples—only requires $O\left(\frac{d}{\epsilon^2}\right)$ samples. This is sample optimal for this problem.

The algorithm is poly-time but relatively slow.
The algorithm is very careful with samples—only requires $O(d/\epsilon^2)$ samples. This is sample optimal for this problem.
The algorithm is very careful with samples—only requires $O\left(\frac{d}{\epsilon^2}\right)$ samples. This is **sample optimal** for this problem.

The algorithm is poly-time but relatively slow.
Unknown covariance

At a high level, similar ideas. Assume $\mu = 0$, so we only need to recover $\Sigma = E[XX^T]$. Similar ideas, but the techniques are more involved...
Unknown covariance

At a high level, similar ideas.
Unknown covariance

At a high level, similar ideas.

Assume $\mu = 0$, so we only need to recover $\Sigma = \mathbb{E}[XX^T]$. 
Unknown covariance

At a high level, similar ideas.

Assume $\mu = 0$, so we only need to recover $\Sigma = \mathbb{E}[XX^T]$.

Previously...

Idea: If the corrupted points move the empirical mean by more than $\tilde{O}(\epsilon)$, then they must also screw up the covariance!
Unknown covariance

At a high level, similar ideas.

Assume $\mu = 0$, so we only need to recover $\Sigma = \mathbb{E}[XX^T]$.

Now:

**Idea:** If the corrupted points move the empirical covariance by more than $\tilde{O}(\epsilon)$, then they must also screw up the covariance of the covariance (i.e. the fourth moment tensor)!

Similar ideas, but the techniques are more involved...
Product distributions

A product distribution over $\{0, 1\}^n$ is a distribution where each coordinate is independent.

Similar techniques allow us to learn the mean to error $\tilde{O}(\epsilon)$ in $\ell_2$.

It almost suffices to learn the mean of the product distribution in $\ell_2$...except when the mean is close to 0 or 1.

Need to use $\chi^2$ distance.

We get $\tilde{O}(\sqrt{\epsilon})$ close in TV distance for general product distributions.
Product distributions

A product distribution over \(\{0, 1\}^n\) is a distribution where each coordinate is independent.
Product distributions

A product distribution over \( \{0, 1\}^n \) is a distribution where each coordinate is independent.

Similar techniques allow us to learn the mean to error \( \tilde{O}(\epsilon) \) in \( \ell_2 \).
Product distributions

A product distribution over \(\{0, 1\}^n\) is a distribution where each coordinate is independent.

Similar techniques allow us to learn the mean to error \(\tilde{O}(\epsilon)\) in \(\ell_2\).

It almost suffices to learn the mean of the product distribution in \(\ell_2\)...except when the mean is close to 0 or 1.
A product distribution over $\{0, 1\}^n$ is a distribution where each coordinate is independent.

Similar techniques allow us to learn the mean to error $\tilde{O}(\epsilon)$ in $\ell_2$.

It almost suffices to learn the mean of the product distribution in $\ell_2$...except when the mean is close to 0 or 1.

Need to use $\chi^2$ distance.
Product distributions

A product distribution over \( \{0, 1\}^n \) is a distribution where each coordinate is independent.

Similar techniques allow us to learn the mean to error \( \tilde{O}(\epsilon) \) in \( \ell_2 \).

It almost suffices to learn the mean of the product distribution in \( \ell_2 \) except when the mean is close to 0 or 1.

Need to use \( \chi^2 \) distance.

We get \( \tilde{O}(\sqrt{\epsilon}) \) close in TV distance for general product distributions.
Mixtures?
Mixtures?

See paper!
We provide the first robust estimators for a wide variety of classes of distributions in high dimensions which are both computationally efficient and provide dimension-independent error guarantees.

Our algorithms use higher moment information at a global level to move beyond naive outlier detection.

We provide two algorithms based on this intuition, which trade off runtime and sample complexity.
Open Questions

- Practicality? Experiments?
- More distributions / more general settings?
- Can we remove the log factors?
- Can we get $O(d/\epsilon^2)$ samples more efficiently?
Open Questions

- Practicality? Experiments?
- More distributions / more general settings?
- Can we remove the log factors?
- Can we get $O(d/\epsilon^2)$ samples more efficiently?

Thank you!