Robust Statistics, Revisited

Ankur Moitra (MIT)

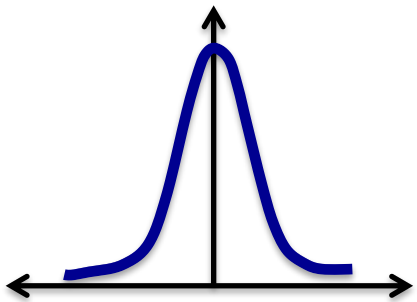
joint work with Ilias Diakonikolas, Jerry Li, Gautam Kamath, Daniel Kane and Alistair Stewart

CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some *class*

e.g. a 1-D Gaussian

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



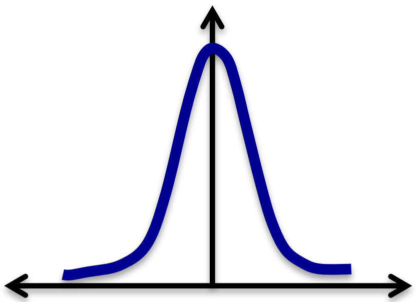
can we accurately estimate its parameters?

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?

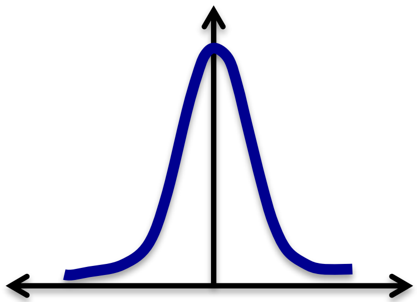
Yes!

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:

$$\frac{1}{N}\sum_{i=1}^{N} (X_i - \overline{X})^2 \to \sigma^2$$

R. A. Fisher

The **maximum likelihood estimator** is asymptotically efficient (1910-1920)

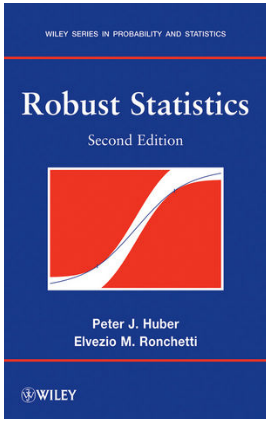
R. A. Fisher

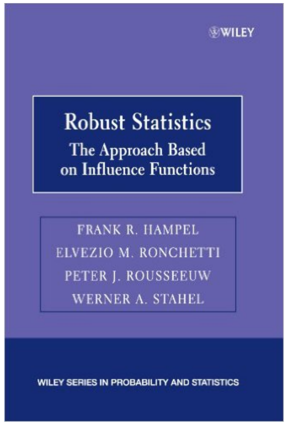
The **maximum likelihood estimator** is asymptotically efficient (1910-1920)

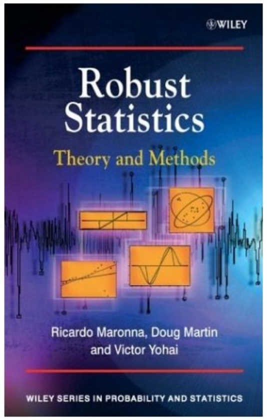
J. W. Tukey

What about **errors** in the model itself? (1960)

ROBUST STATISTICS

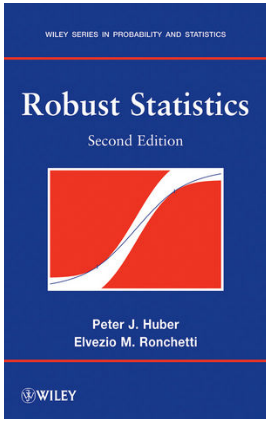


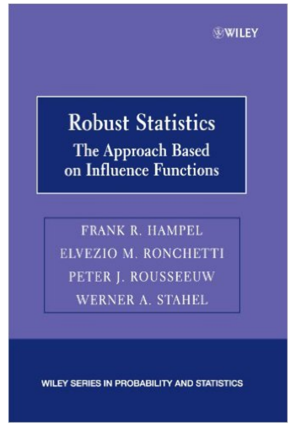


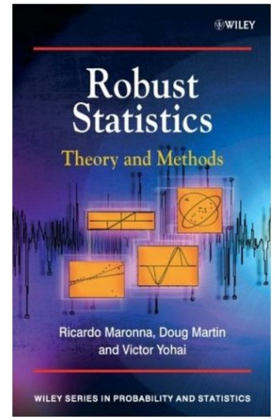


What estimators behave well in a **neighborhood** around the model?

ROBUST STATISTICS





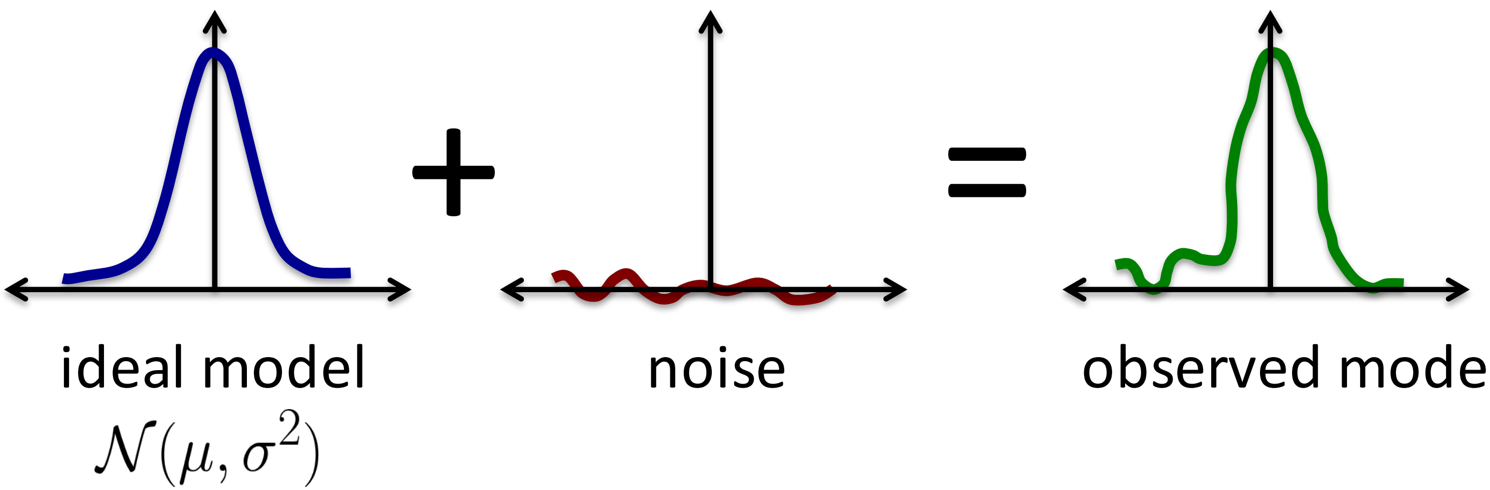


What estimators behave well in a **neighborhood** around the model?

Let's study a simple one-dimensional example….

ROBUST PARAMETER ESTIMATION

Given **corrupted** samples from a 1-D Gaussian:



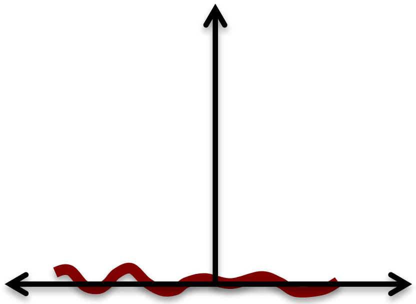
can we accurately estimate its parameters?

How do we constrain the noise?

How do we constrain the noise?

Equivalently:

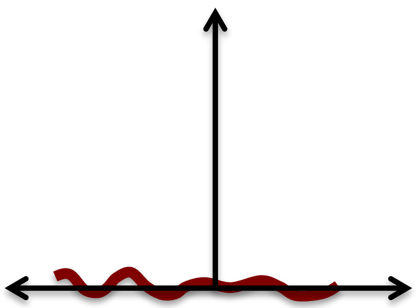
L_1-norm of noise at most $O(\varepsilon)$



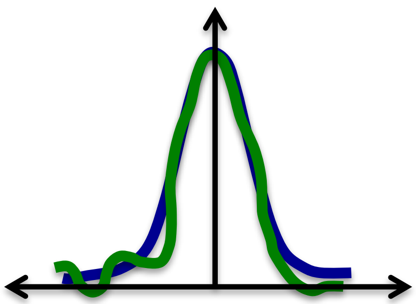
How do we constrain the noise?

Equivalently:

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



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

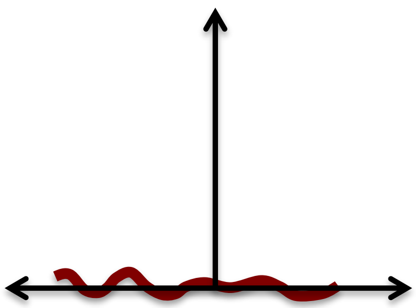


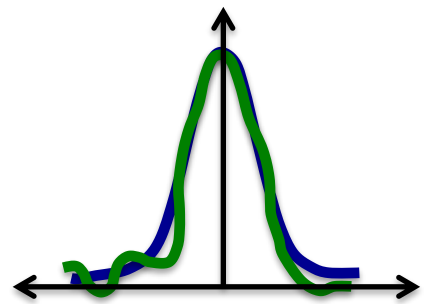
How do we constrain the noise?

Equivalently:

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

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



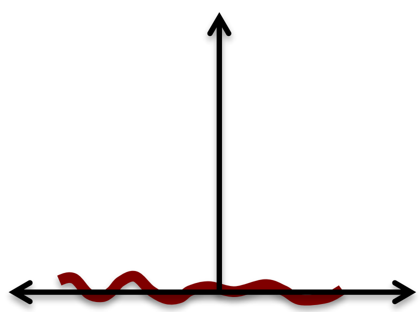


This generalizes **Huber's Contamination Model:** An adversary can add an ε-fraction of samples

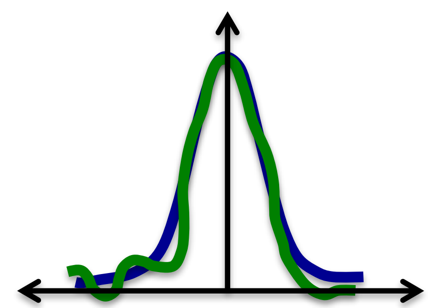
How do we constrain the noise?

Equivalently:

L_1-norm of noise at most O(ε)



Arbitrarily corrupt O(ε)-fraction of samples (in expectation)



This generalizes **Huber's Contamination Model:** An adversary can add an ε-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

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

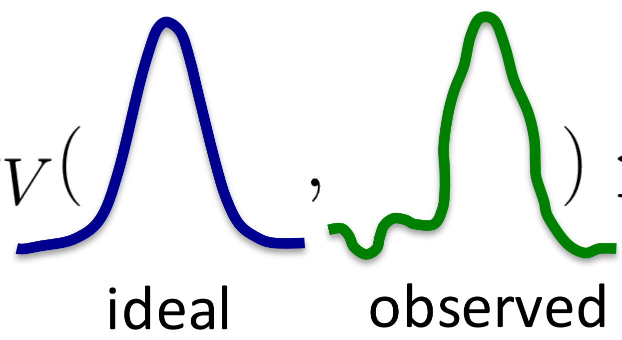
In what norm do we want the parameters to be close?

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

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

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

$$d_{TV}(\quad , \quad) \leq O(\epsilon)$$

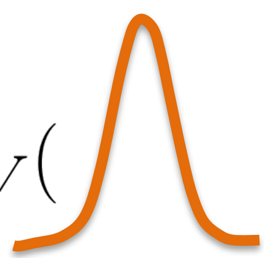
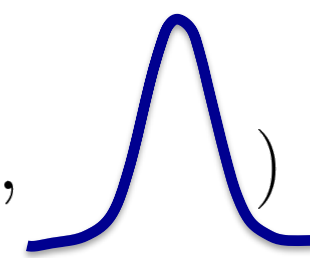


In what norm do we want the parameters to be close?

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

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

Goal: Find a 1-D Gaussian that satisfies

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

estimate

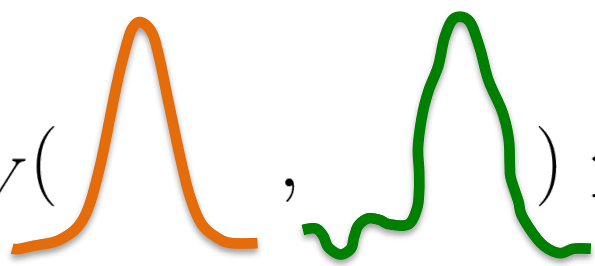
ideal

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

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

Equivalently, find a 1-D Gaussian that satisfies

$d_{TV}($  $) \leq O(\epsilon)$

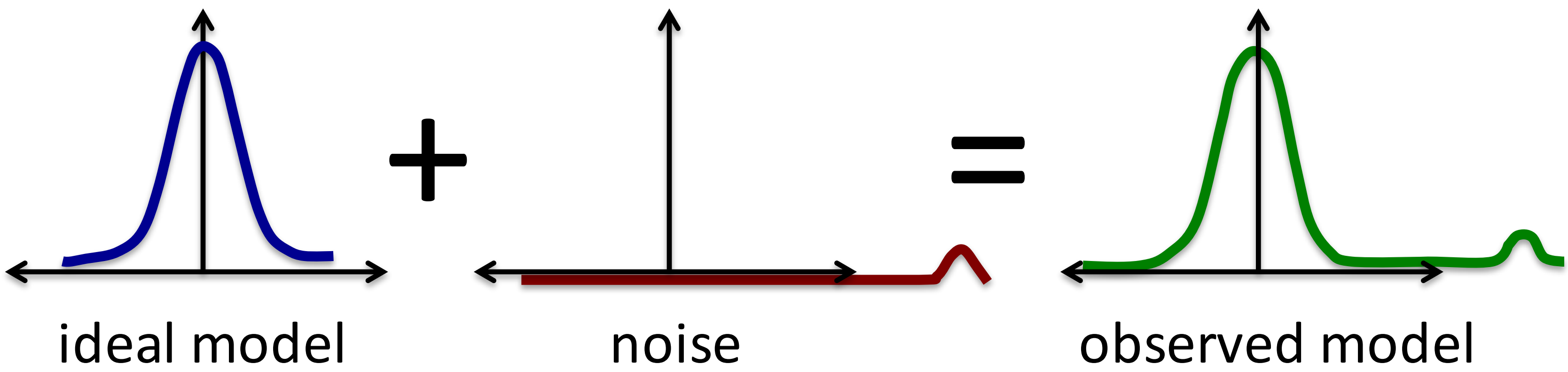
Do the empirical mean and empirical variance work?

Do the empirical mean and empirical variance work?

No!

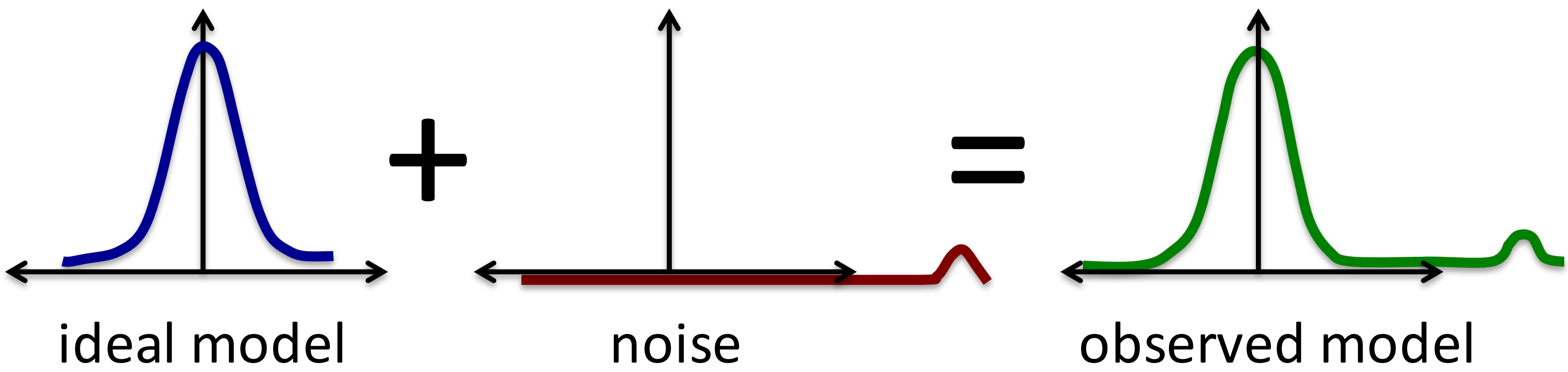
Do the empirical mean and empirical variance work?

No!



Do the empirical mean and empirical variance work?

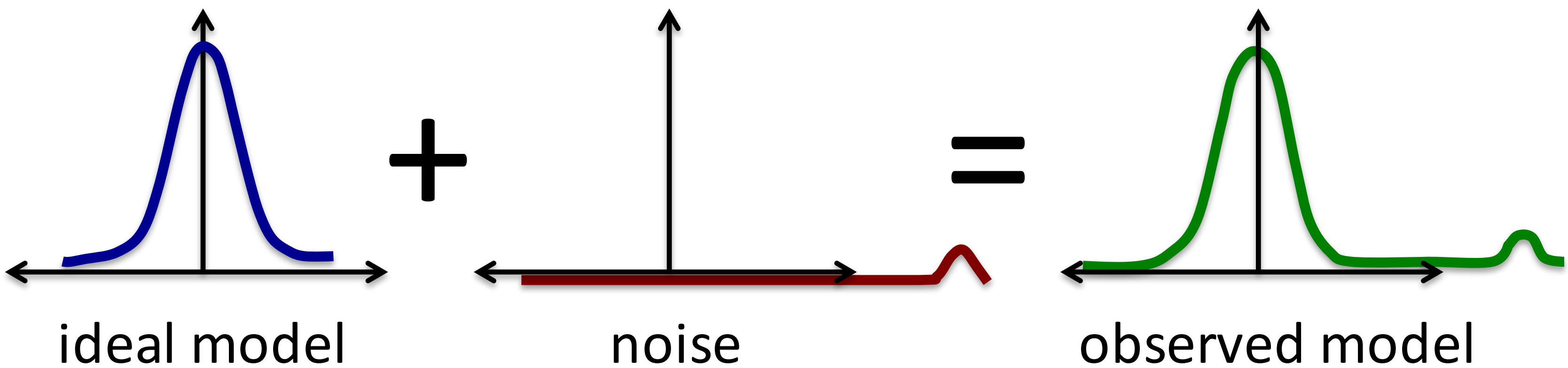
No!



A single corrupted sample can arbitrarily corrupt the estimates

Do the empirical mean and empirical variance work?

No!

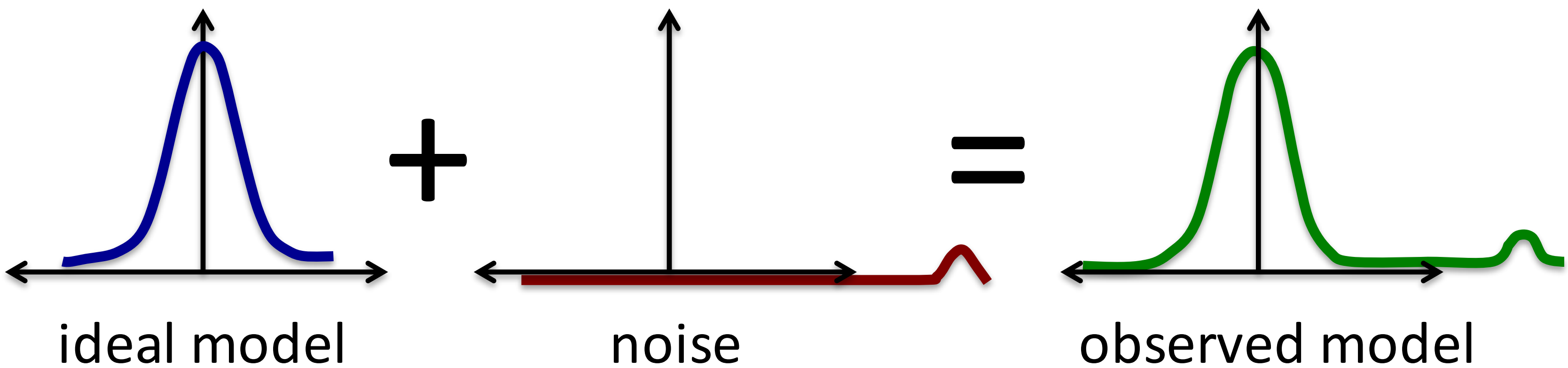


A single corrupted sample can arbitrarily corrupt the estimates

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

Do the empirical mean and empirical variance work?

No!



A single corrupted sample can arbitrarily corrupt the estimates

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

$$\mathrm{MAD} = \mathrm{median}(|X_i - \mathrm{median}(X_1, X_2, \ldots, X_n)|)$$

Fact [Folklore]: Given samples from a distribution that are ε-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

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

where $\widehat{\mu} = \mathrm{median}(X), \; \widehat{\sigma} = \dfrac{\mathrm{MAD}}{\Phi^{-1}(3/4)}$

Fact [Folklore]: Given samples from a distribution that are ε-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

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

where $\widehat{\mu} = \mathrm{median}(X), \ \widehat{\sigma} = \dfrac{\mathrm{MAD}}{\Phi^{-1}(3/4)}$

Also called (properly) **agnostically learning** a 1-D Gaussian

Fact [Folklore]: Given samples from a distribution that are ε-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

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

where $\widehat{\mu} = \mathrm{median}(X), \ \widehat{\sigma} = \dfrac{\mathrm{MAD}}{\Phi^{-1}(3/4)}$

What about robust estimation in high-dimensions?

Fact [Folklore]: Given samples from a distribution that are ε-close in total variation distance to a 1-D Gaussian

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

the median and MAD recover estimates that satisfy

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

where $\widehat{\mu} = \mathrm{median}(X), \ \widehat{\sigma} = \dfrac{\mathrm{MAD}}{\Phi^{-1}(3/4)}$

What about robust estimation in high-dimensions?

e.g. microarrays with 10k genes

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

Main Problem: Given samples from a distribution that are ε-close in total variation distance to a d-dimensional Gaussian

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

give an efficient algorithm to find parameters that satisfy

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

Main Problem: Given samples from a distribution that are ε-close in total variation distance to a d-dimensional Gaussian

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

give an efficient algorithm to find parameters that satisfy

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

Special Cases:

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

(2) Unknown covariance $\mathcal{N}(0, \Sigma)$

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median		

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	O(ε) ✓	

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	O(ε) ✓	NP-Hard ✗

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	O(ε) ✓	NP-Hard ✗
Geometric Median		

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	O(ε) ✓	NP-Hard ✗
Geometric Median		poly(d,N) ✓

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	$O(\varepsilon)$ ✓	NP-Hard ✗
Geometric Median	$O(\varepsilon\sqrt{d})$ ✗	poly(d,N) ✓

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	$O(\varepsilon)$	NP-Hard
Geometric Median	$O(\varepsilon\sqrt{d})$	poly(d,N)
Tournament	$O(\varepsilon)$	$N^{O(d)}$

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	$O(\varepsilon)$ ✓	NP-Hard ✗
Geometric Median	$O(\varepsilon\sqrt{d})$ ✗	poly(d,N) ✓
Tournament	$O(\varepsilon)$ ✓	$N^{O(d)}$ ✗
Pruning	$O(\varepsilon\sqrt{d})$ ✗	$O(dN)$ ✓

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	$O(\varepsilon)$ ✓	NP-Hard ✗
Geometric Median	$O(\varepsilon\sqrt{d})$ ✗	poly(d,N) ✓
Tournament	$O(\varepsilon)$ ✓	$N^{O(d)}$ ✗
Pruning	$O(\varepsilon\sqrt{d})$ ✗	$O(dN)$ ✓

⋮

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?

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUR RESULTS

Robust estimation is high-dimensions is algorithmically possible!

Theorem [Diakonikolas, Li, Kamath, Kane, Moitra, Stewart '16]:
There is an algorithm when given $N = \widetilde{O}(d^3/\epsilon^2)$ samples from a distribution that is ε-close in total variation distance to a d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

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

Moreover the algorithm runs in time poly(N, d)

OUR RESULTS

Robust estimation is high-dimensions is algorithmically possible!

Theorem [Diakonikolas, Li, Kamath, Kane, Moitra, Stewart '16]: There is an algorithm when given $N = \widetilde{O}(d^3/\epsilon^2)$ samples from a distribution that is ε-close in total variation distance to a d-dimensional Gaussian $\mathcal{N}(\mu, \Sigma)$ finds parameters that satisfy

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

Moreover the algorithm runs in time poly(N, d)

Alternatively: Can approximate the Tukey median, etc, in interesting semi-random models

Simultaneously **[Lai, Rao, Vempala '16]** gave agnostic algorithms that achieve:

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

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

and work for non-Gaussian distributions too

Simultaneously **[Lai, Rao, Vempala '16]** gave agnostic algorithms that achieve:

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

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

and work for non-Gaussian distributions too

Many other applications across both papers: product distributions, mixtures of spherical Gaussians, SVD, ICA

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**…

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- **Parameter Distance**
- Detecting When an Estimator is Compromised
- Filtering and Convex Programming
- Unknown Covariance

Part III: Experiments and Extensions

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:

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

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

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

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

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

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

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

$$\|\mu - \widehat{\mu}\|_2 \leq \widetilde{O}(\epsilon)$$

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

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

A Basic Fact:

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

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

$$\|\mu - \widehat{\mu}\|_2 \leq \widetilde{O}(\epsilon)$$

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

Our new goal is to be close in **Euclidean distance**

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- **Detecting When an Estimator is Compromised**
- Filtering and Convex Programming
- Unknown Covariance

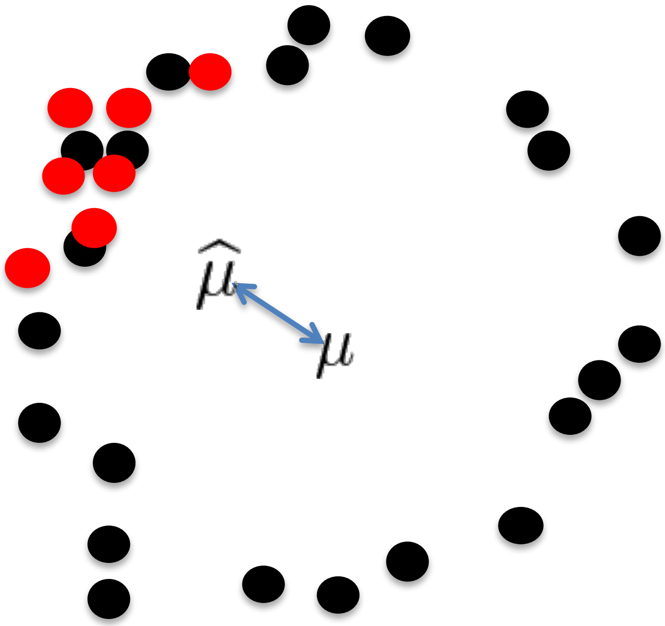
Part III: Experiments and Extensions

DETECTING CORRUPTIONS

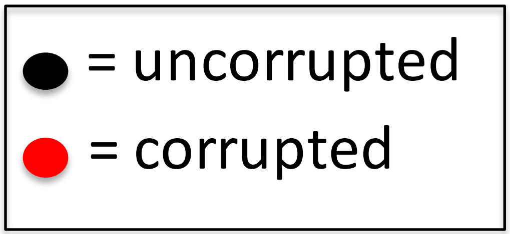
Step #2: Detect when the naïve estimator has been compromised

DETECTING CORRUPTIONS

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

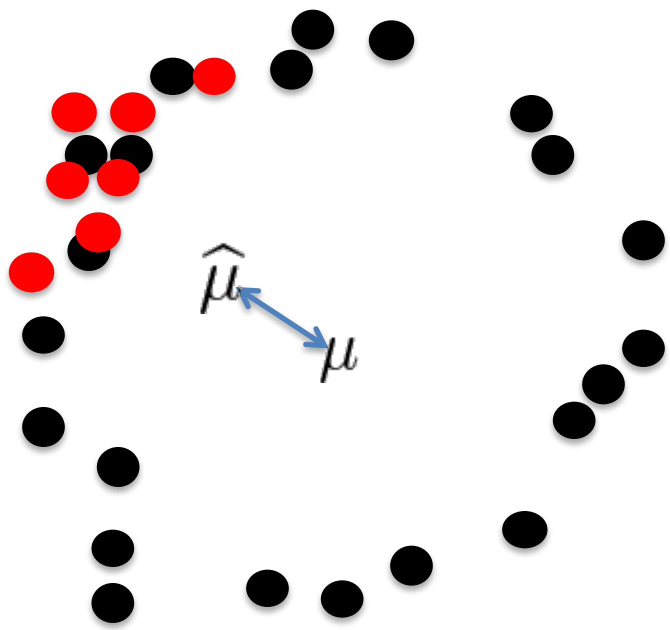


$$\widehat{\mu} \triangleq \frac{1}{N} \sum_{i=1}^{N} X_i$$

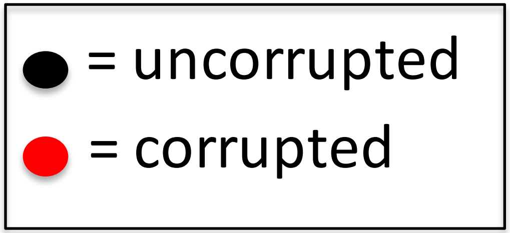


DETECTING CORRUPTIONS

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



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



There is a direction of large (> 1) variance

Key Lemma: If X_1, X_2, … X_N come from a distribution that is ε-close to $\mathcal{N}(\mu, I)$ and $N \geq 10(d + \log 1/\delta)/\epsilon^2$ then for

(1) $\widehat{\mu} \triangleq \frac{1}{N}\sum_{i=1}^{N} X_i$ (2) $\widehat{\Sigma} \triangleq \frac{1}{N}\sum_{i=1}^{N}(X_i - \widehat{\mu})(X_i - \widehat{\mu})^T$

with probability at least 1-δ

$$\|\mu - \widehat{\mu}\|_2 \geq C\epsilon\sqrt{\log 1/\epsilon} \longrightarrow \|\widehat{\Sigma} - I\|_2 \geq C'\epsilon \log 1/\epsilon$$

Key Lemma: If X_1, X_2, … X_N come from a distribution that is ε-close to $\mathcal{N}(\mu, I)$ and $N \geq 10(d + \log 1/\delta)/\epsilon^2$ then for

$$\text{(1)} \quad \widehat{\mu} \triangleq \frac{1}{N}\sum_{i=1}^{N} X_i \qquad \text{(2)} \quad \widehat{\Sigma} \triangleq \frac{1}{N}\sum_{i=1}^{N}(X_i - \widehat{\mu})(X_i - \widehat{\mu})^T$$

with probability at least 1-δ

$$\|\mu - \widehat{\mu}\|_2 \geq C\epsilon\sqrt{\log 1/\epsilon} \longrightarrow \|\widehat{\Sigma} - I\|_2 \geq C'\epsilon \log 1/\epsilon$$

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

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- **Filtering and Convex Programming**
- Unknown Covariance

Part III: Experiments and Extensions

OUR ALGORITHM(S)

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

OUR ALGORITHM(S)

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

Filtering Approach: Suppose that:

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

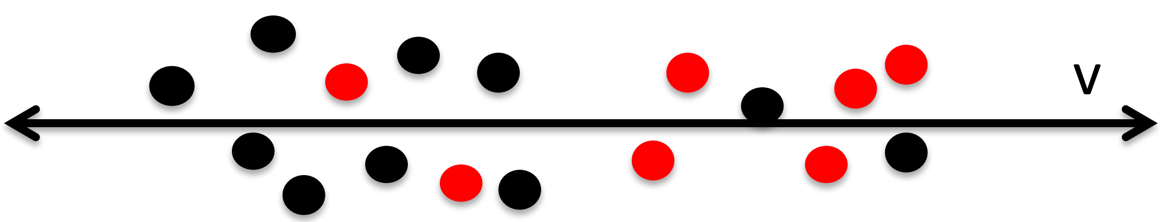
OUR ALGORITHM(S)

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

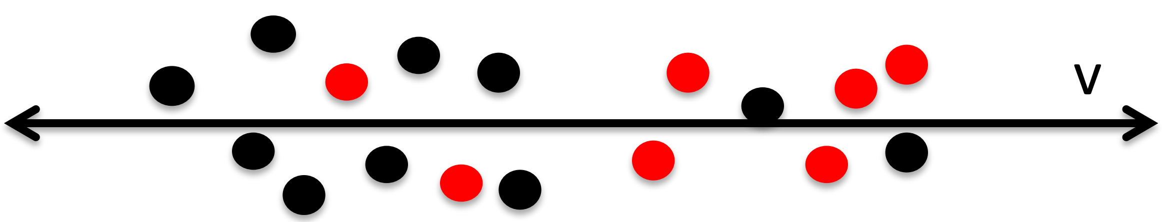
OUR ALGORITHM(S)

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

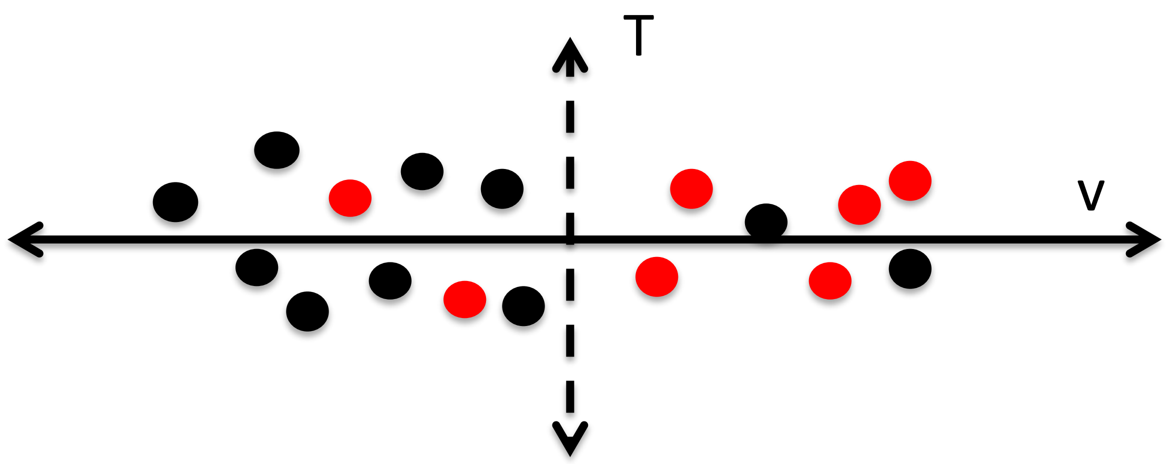
OUR ALGORITHM(S)

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

OUR ALGORITHM(S)

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

OUR ALGORITHM(S)

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

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

OUR ALGORITHM(S)

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points

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

Eventually we find (certifiably) good parameters

OUR ALGORITHM(S)

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

Filtering Approach: Suppose that:

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

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: $\widetilde{O}(Nd^2)$ **Sample Complexity:** $\widetilde{O}(d^2/\epsilon^2)$

OUR ALGORITHM(S)

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

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

Eventually we find (certifiably) good parameters

Running Time: $\widetilde{O}(Nd^2)$ **Sample Complexity:** $\widetilde{O}(d^2/\epsilon^2)$

Concentration of LTFs

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

- Parameter Distance
- Detecting When an Estimator is Compromised
- Filtering and Convex Programming
- **Unknown Covariance**

Part III: Experiments and Extensions

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

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

Another Basic Fact:

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

Again, proven using Pinsker's Inequality

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

Another Basic Fact:

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

Again, proven using Pinsker's Inequality

Our new goal is to find an estimate that satisfies:

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

PARAMETER DISTANCE

Step #1: Find an appropriate parameter distance for Gaussians

Another Basic Fact:

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

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?

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

UNKNOWN COVARIANCE

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

How do we detect if the naïve estimator is compromised?

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

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 x d symmetric matrices

$$M = 2\Sigma^{\otimes 2} + \left(\Sigma^{\flat}\right)\left(\Sigma^{\flat}\right)^T$$

UNKNOWN COVARIANCE

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

How do we detect if the naïve estimator is compromised?

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

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 x d symmetric matrices

$$M = 2\Sigma^{\otimes 2} + \left(\Sigma^{\flat}\right)\left(\Sigma^{\flat}\right)^T$$

Proof uses **Isserlis'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?

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

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 x d symmetric matrices

$$M = 2\Sigma^{\otimes 2} + \underbrace{\left(\Sigma^{\flat}\right)\left(\Sigma^{\flat}\right)^T}_{\text{need to project out}}$$

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

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

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

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

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:

$$\frac{1}{N}\sum_{i=1}^{N} \Big(Y_i \otimes Y_i\Big)\Big(Y_i \otimes Y_i\Big)^T - 2I$$

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:

$$\frac{1}{N}\sum_{i=1}^{N}\Big(Y_i \otimes Y_i\Big)\Big(Y_i \otimes Y_i\Big)^T - 2I$$

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

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 $X_i - X_i' \sim_\epsilon \mathcal{N}(0, 2\Sigma)$

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**

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

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

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

$$\widehat{\Sigma}^{-1/2} X_i \sim_\epsilon \mathcal{N}(\underbrace{\widehat{\Sigma}^{-1/2}\mu}, I)$$

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

$$\widehat{\Sigma}^{-1/2} X_i \sim_\epsilon \mathcal{N}(\underbrace{\widehat{\Sigma}^{-1/2}\mu}_{\text{right distance, in general case}}, I)$$

Now use algorithm for **unknown mean**

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

OUTLINE

Part I: Introduction

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

Part II: Agnostically Learning a Gaussian

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

Part III: Experiments and Extensions

FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

Binary Product Distributions:

$$d_{TV}(\Pi, \widehat{\Pi}) \leq C\sqrt{\epsilon \log 1/\epsilon}$$

FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

Binary Product Distributions:

$$d_{TV}(\Pi, \widehat{\Pi}) \leq C\sqrt{\epsilon \log 1/\epsilon}$$

Mixtures of Two c-Balanced Binary Product Distributions:

$$d_{TV}(\Pi, \widehat{\Pi}) \leq C\epsilon^{1/6}$$

FURTHER RESULTS

Use restricted eigenvalue problems to detect outliers

Binary Product Distributions:

$$d_{TV}(\Pi, \widehat{\Pi}) \leq C\sqrt{\epsilon \log 1/\epsilon}$$

Mixtures of Two c-Balanced Binary Product Distributions:

$$d_{TV}(\Pi, \widehat{\Pi}) \leq C\epsilon^{1/6}$$

Mixtures of k Spherical Gaussians:

$$d_{TV}(\mathcal{M}, \widehat{\mathcal{M}}) \leq C \ \mathrm{poly}(k)\sqrt{\epsilon} \log 1/\epsilon$$

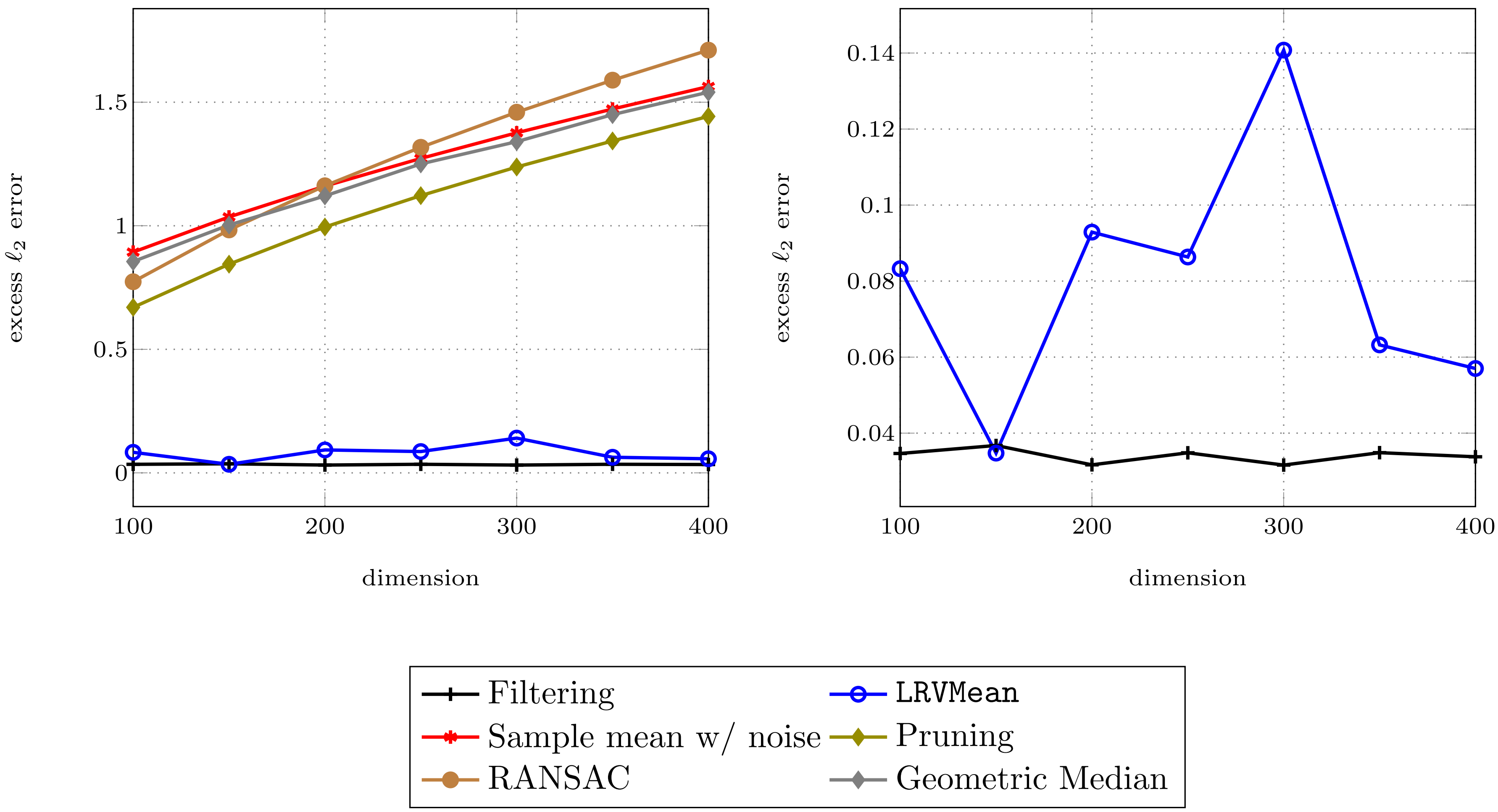
SYNTHETIC EXPERIMENTS

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

$\mathcal{N}(\mu, I)$ **+ 10% noise**

SYNTHETIC EXPERIMENTS

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



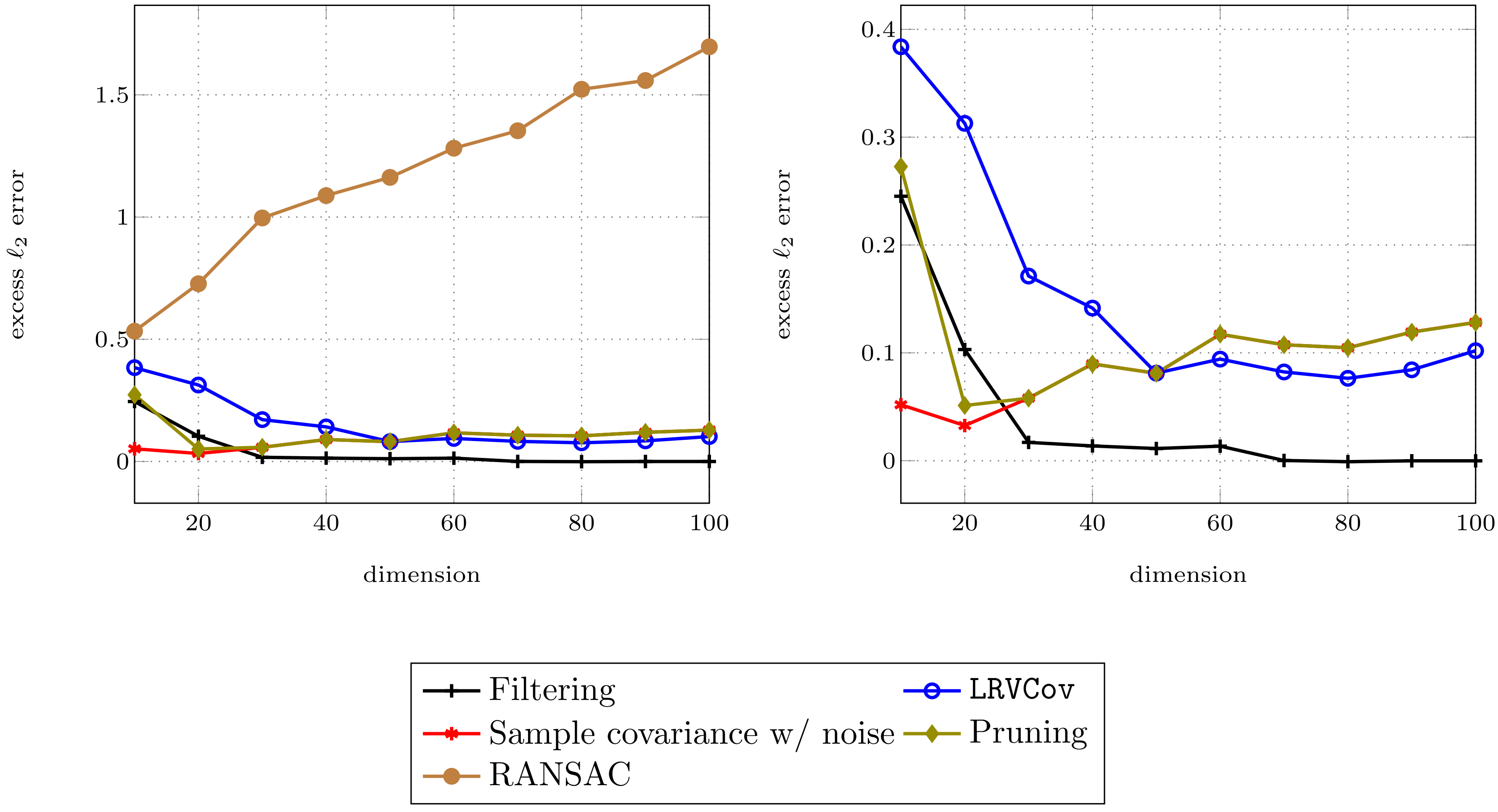
SYNTHETIC EXPERIMENTS

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

$$\mathcal{N}(0, \underbrace{\Sigma}_{\text{close to identity}}) \textbf{ + 10\% noise}$$

SYNTHETIC EXPERIMENTS

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



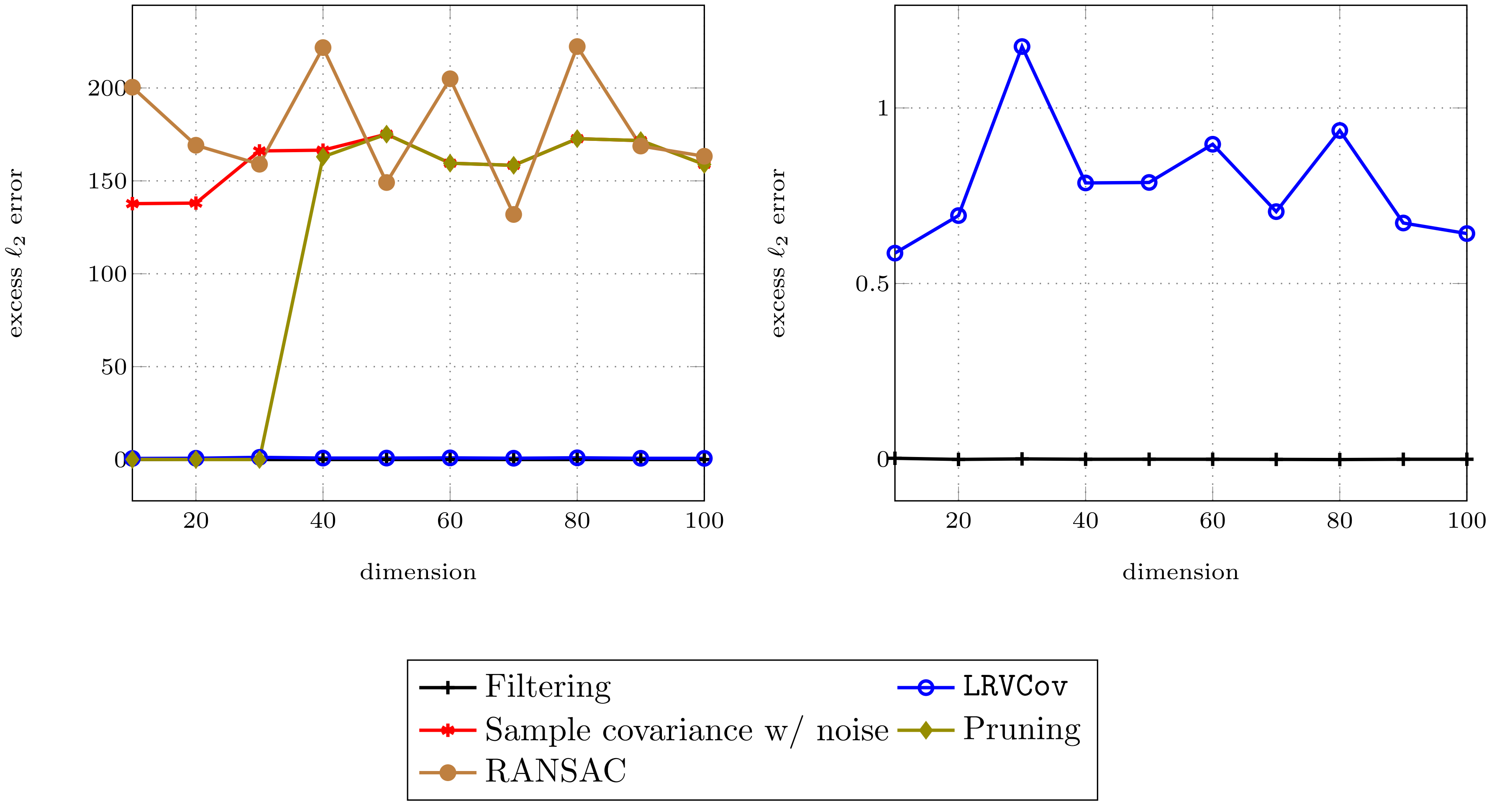
SYNTHETIC EXPERIMENTS

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

$$\mathcal{N}(0, \underbrace{\Sigma}_{\text{far from identity}})$$ **+ 10% noise**

SYNTHETIC EXPERIMENTS

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



REAL DATA EXPERIMENTS

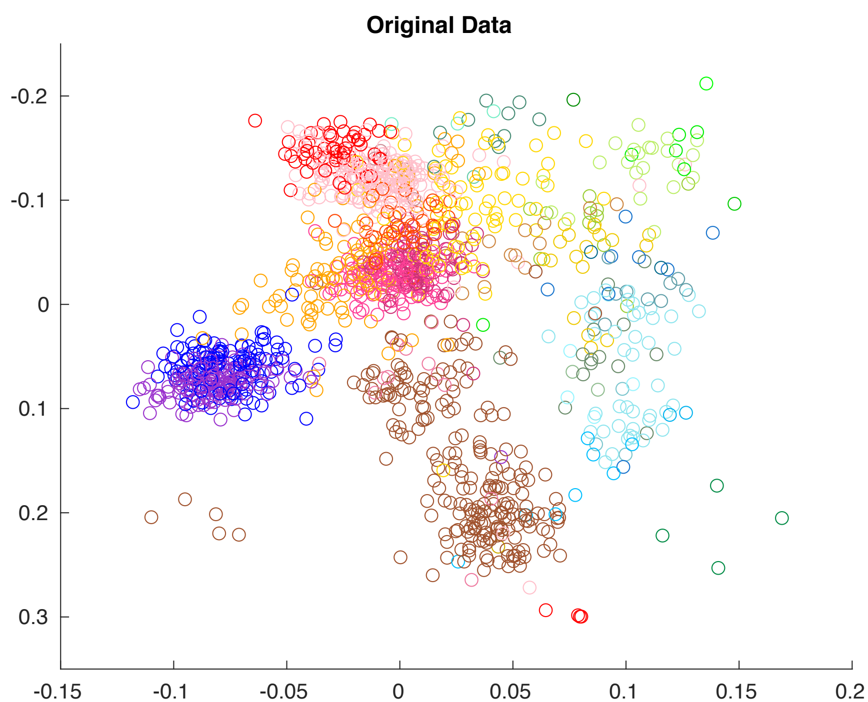
Famous study of **[Novembre et al. '08]**: Take top two singular vectors of people x SNP matrix (POPRES)

REAL DATA EXPERIMENTS

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

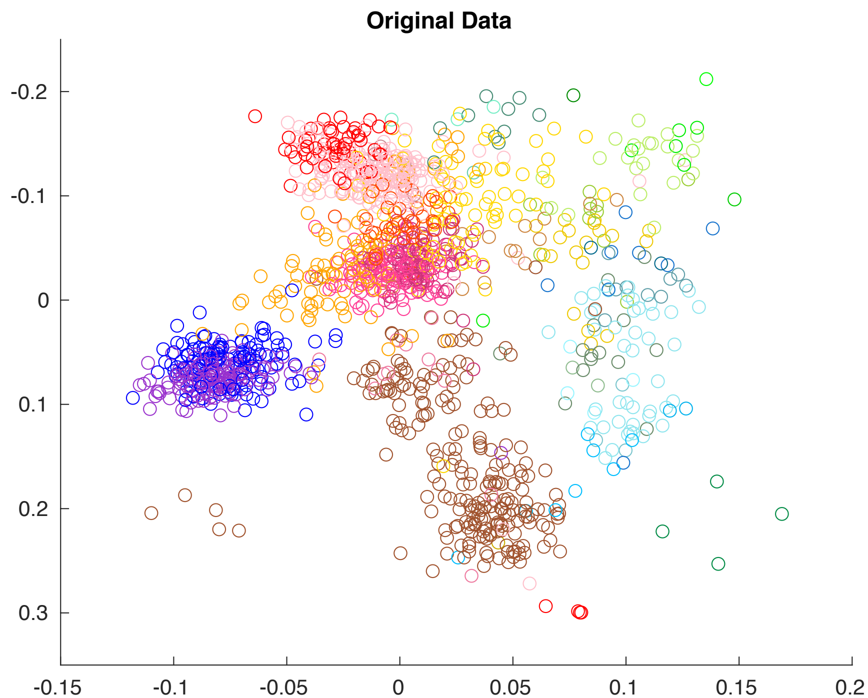
REAL DATA EXPERIMENTS

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"

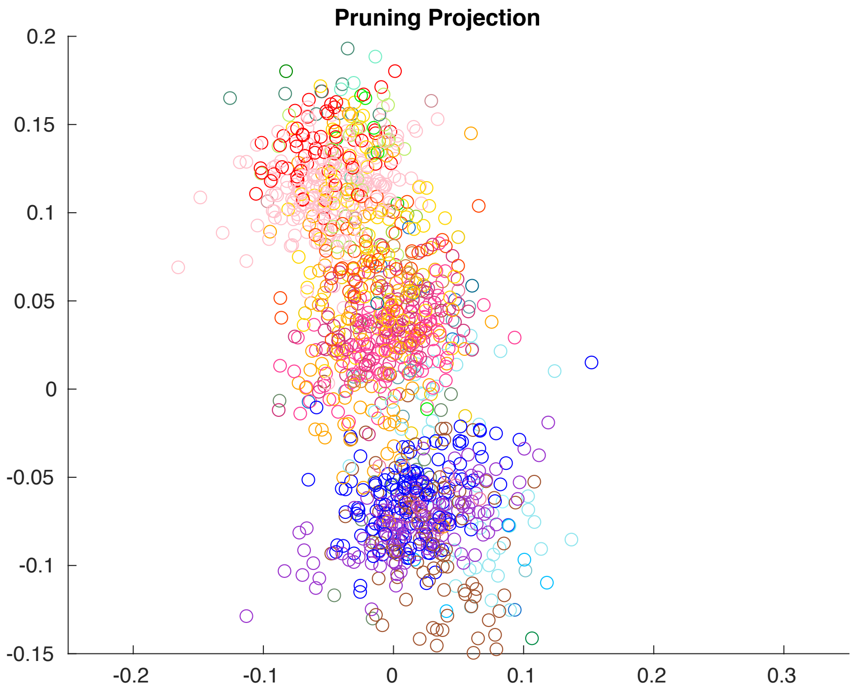
REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

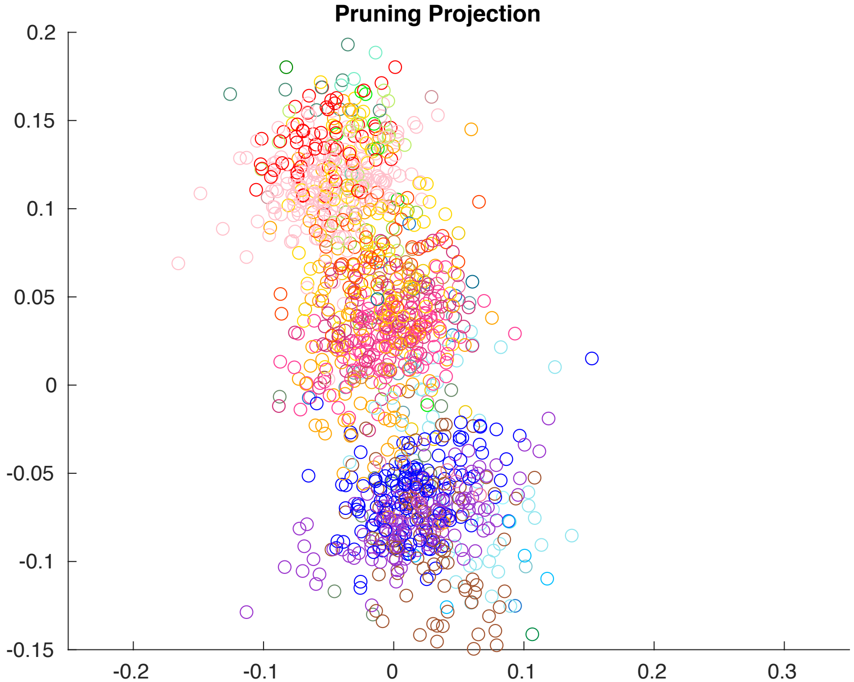


What PCA finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

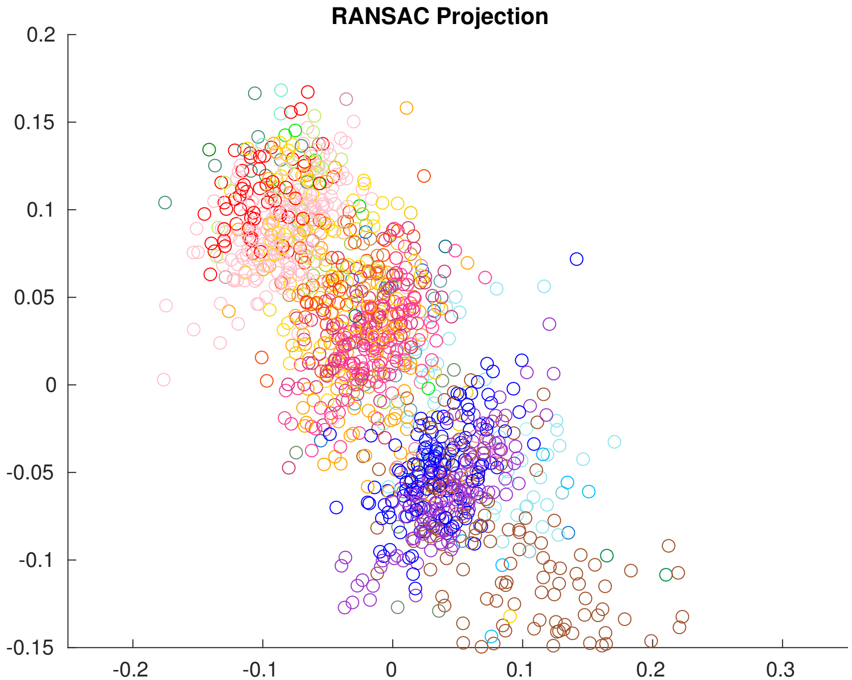


What PCA finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

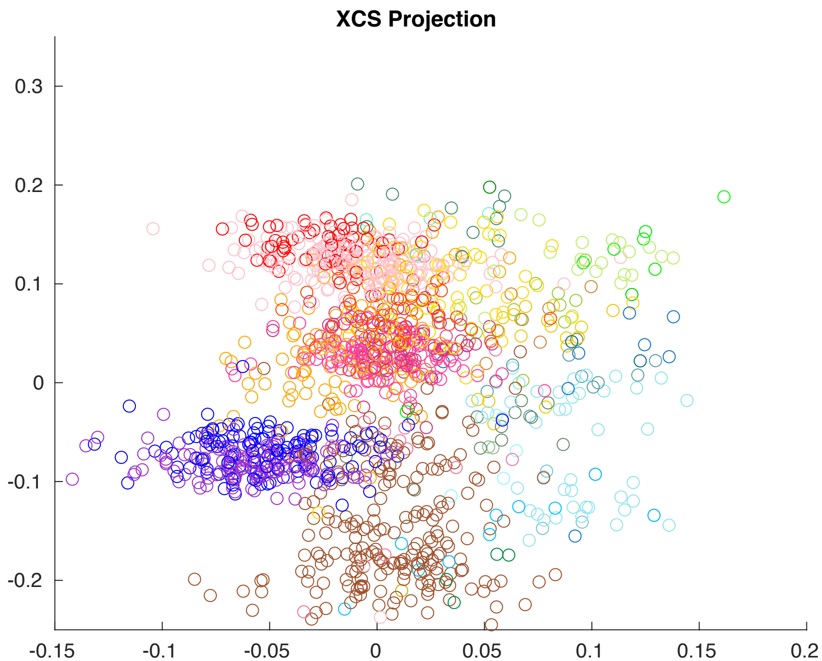


What RANSAC finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

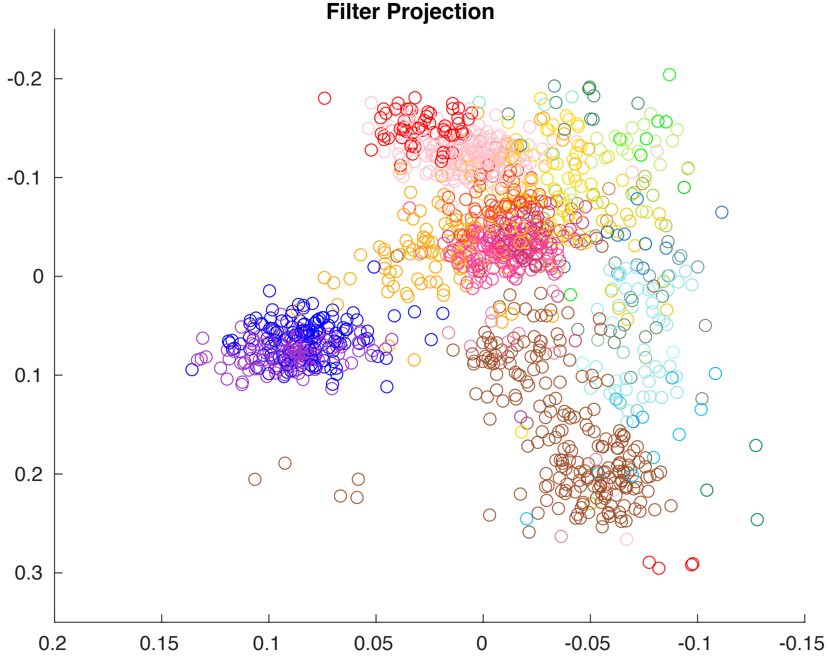


What robust PCA (via SDPs) finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

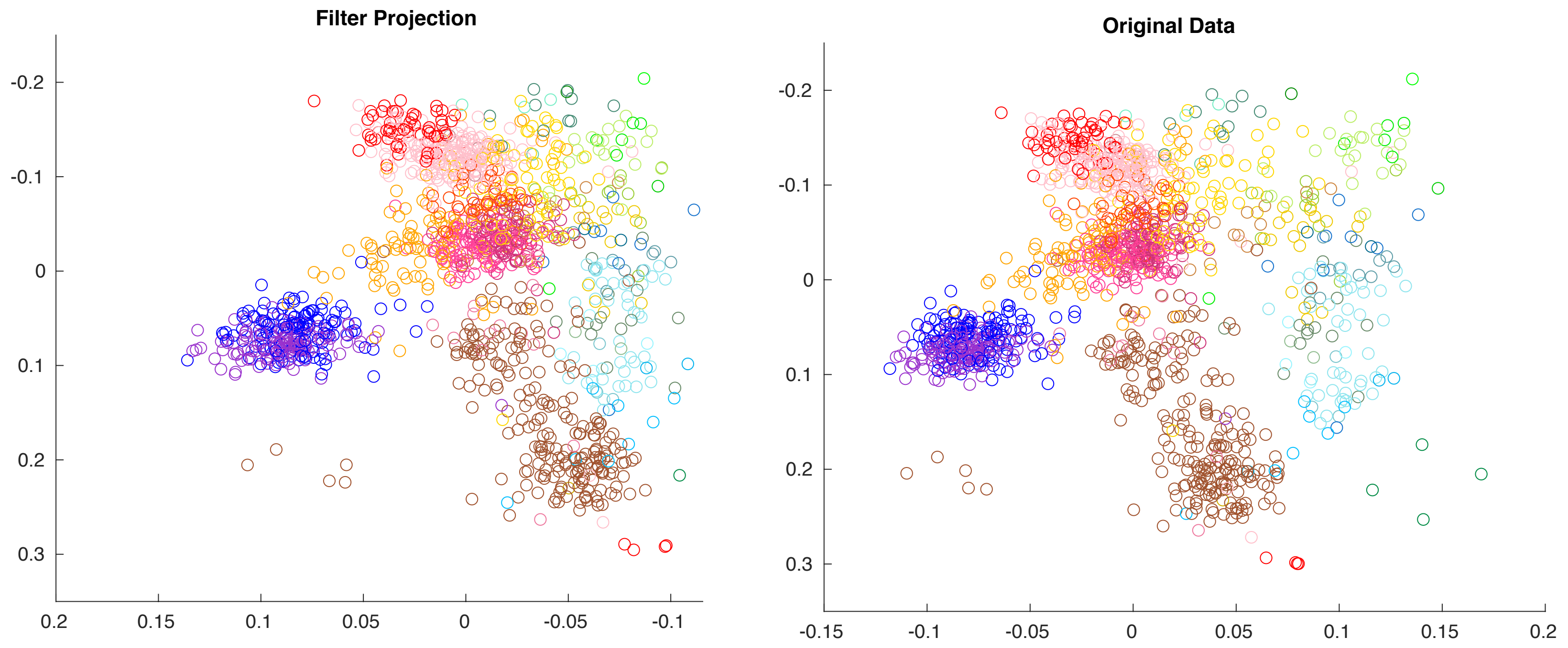


What our methods find

REAL DATA EXPERIMENTS

The power of provably robust estimation:

10% noise **no noise**



What our methods find

LOOKING FORWARD

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

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?

Summary:

- Nearly optimal algorithm for agnostically learning a high-dimensional Gaussian
- General recipe using restricted eigenvalue problems
- Further applications to other mixture models
- **Is practical, robust statistics within reach?**

Thanks! Any Questions?