Robust Statistics, Adversaries and Algorithms

Ankur Moitra (MIT)

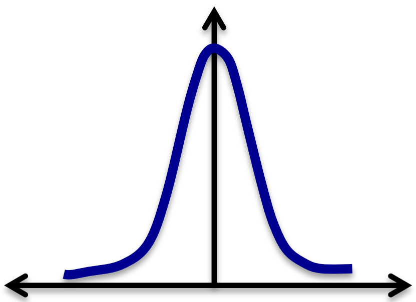
6.S979: Topics in Deployable ML, September 19^{th}

CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some *class*

e.g. a 1-D Gaussian

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



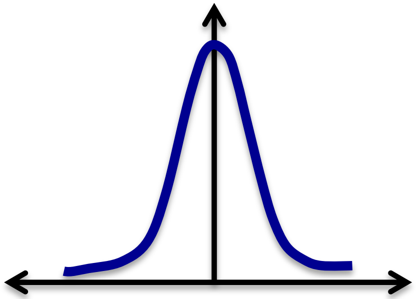
can we accurately estimate its parameters?

CLASSIC PARAMETER ESTIMATION

Given samples from an unknown distribution in some *class*

e.g. a 1-D Gaussian

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



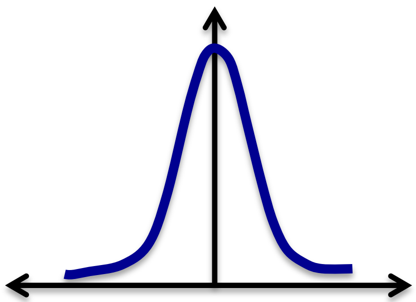
can we accurately estimate its parameters? Yes!

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

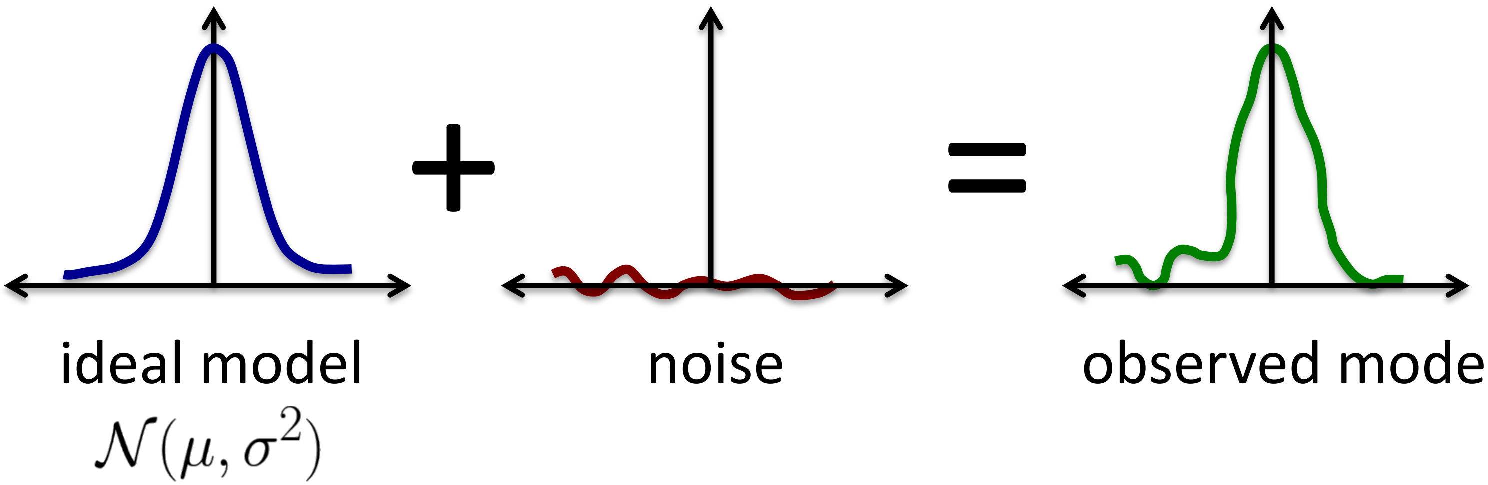
J. W. Tukey

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

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

ROBUST PARAMETER ESTIMATION

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



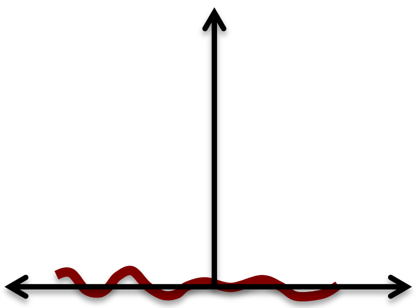
can we accurately estimate its parameters?

How do we constrain the noise?

How do we constrain the noise?

Equivalently:

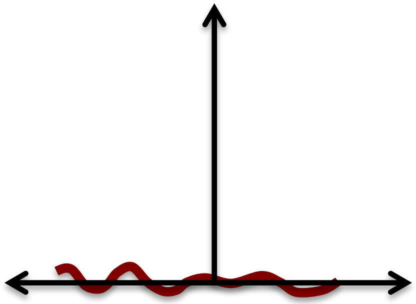
L_1-norm of noise at most $O(\varepsilon)$



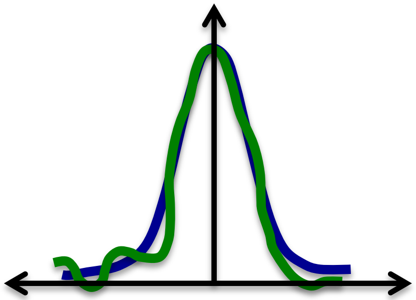
How do we constrain the noise?

Equivalently:

L_1-norm of noise at most O(ε)



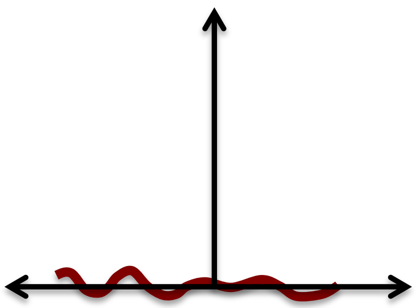
Arbitrarily corrupt O(ε)-fraction of samples (in expectation)



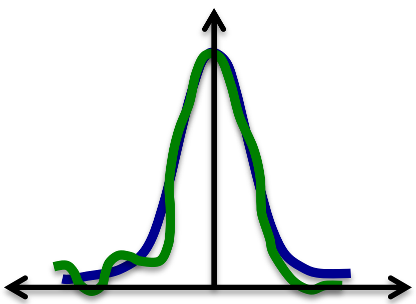
How do we constrain the noise?

Equivalently:

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



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

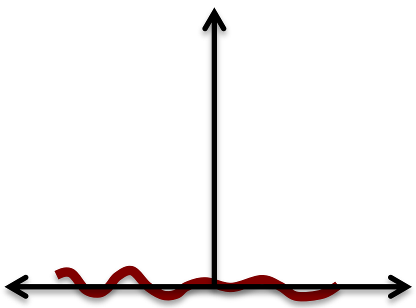


This generalizes **Huber's Contamination Model:** An adversary can add an ε-fraction of samples

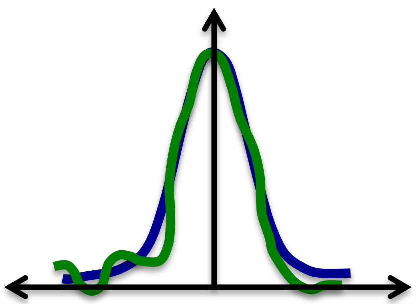
How do we constrain the noise?

Equivalently:

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



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



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

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

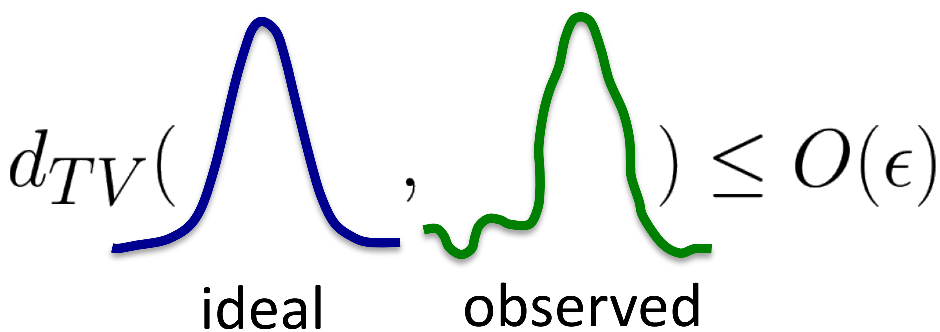
$$d_{TV}(f(x), g(x)) \triangleq \frac{1}{2}\int_{-\infty}^{\infty} \Big| f(x) - g(x) \Big| dx$$

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

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

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

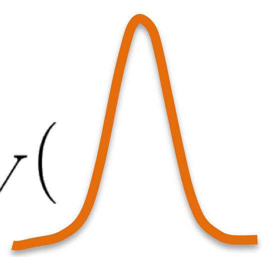
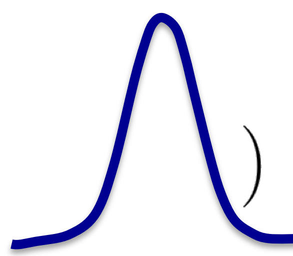


In what norm do we want the parameters to be close?

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

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

Goal: Find a 1-D Gaussian that satisfies

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

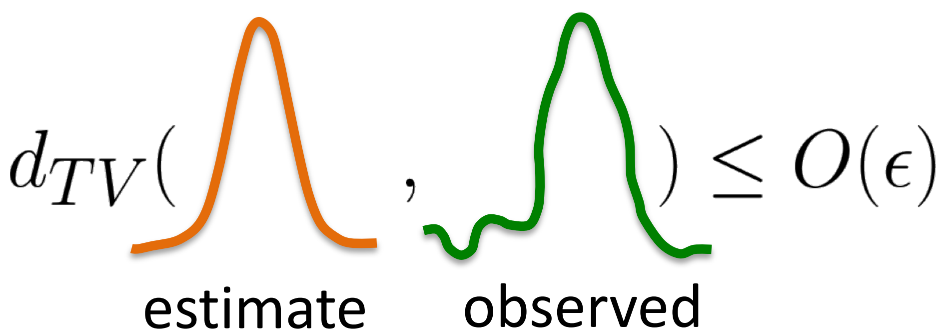
estimate ideal

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



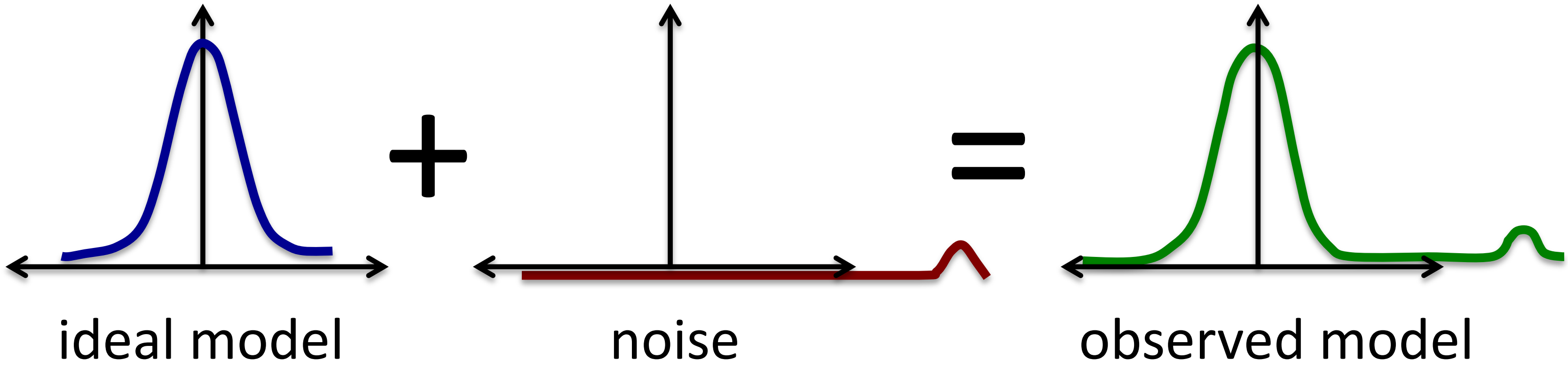
Do the empirical mean and empirical variance work?

Do the empirical mean and empirical variance work?

No!

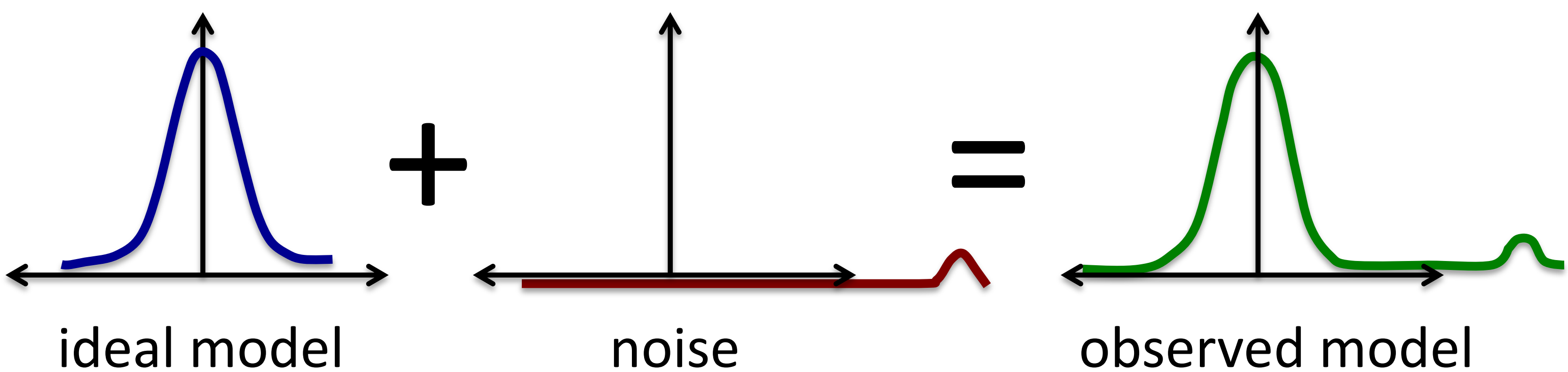
Do the empirical mean and empirical variance work?

No!



Do the empirical mean and empirical variance work?

No!



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 is 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 is 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 is 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 is 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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

Main Problem: Given samples from a distribution that is 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 is 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	$O(\varepsilon)$ ✓	NP-Hard ✗
Geometric Median		

A COMPENDIUM OF APPROACHES

Unknown Mean	Error Guarantee	Running Time
Tukey Median	$O(\varepsilon)$ ✓	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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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)

Extensions: Can weaken assumptions to sub-Gaussian or bounded second moments (with weaker guarantees) for the mean

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

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

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

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

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

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

When the covariance is bounded, this translates to:

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

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

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

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

When the covariance is bounded, this translates to:

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

Subsequently many works **handling more errors via list decoding**,

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

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

When the covariance is bounded, this translates to:

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

Subsequently many works **handling more errors via list decoding**, **giving lower bounds against statistical query algorithms**,

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

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

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

When the covariance is bounded, this translates to:

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

Subsequently many works **handling more errors via list decoding**, **giving lower bounds against statistical query algorithms**, **weakening the distributional assumptions**,

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

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

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

When the covariance is bounded, this translates to:

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

Subsequently many works **handling more errors via list decoding**, **giving lower bounds against statistical query algorithms**, **weakening the distributional assumptions**, **exploiting sparsity**,

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

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

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

When the covariance is bounded, this translates to:

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

Subsequently many works **handling more errors via list decoding**, **giving lower bounds against statistical query algorithms**, **weakening the distributional assumptions**, **exploiting sparsity**, **working with more complex generative models**

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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**
- A Win-Win Algorithm
- Unknown Covariance

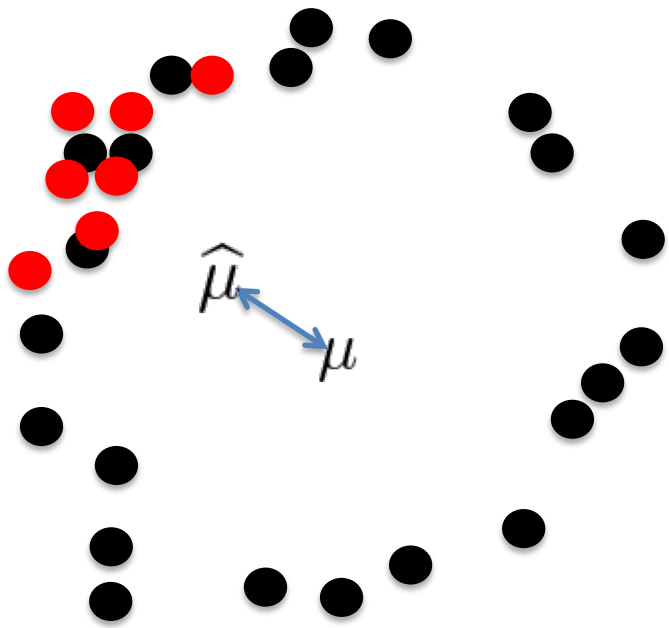
Part III: Experiments

DETECTING CORRUPTIONS

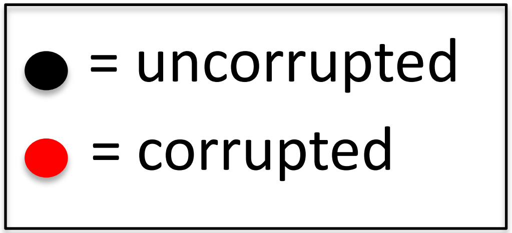
Step #2: Detect when the naïve estimator has been compromised

DETECTING CORRUPTIONS

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

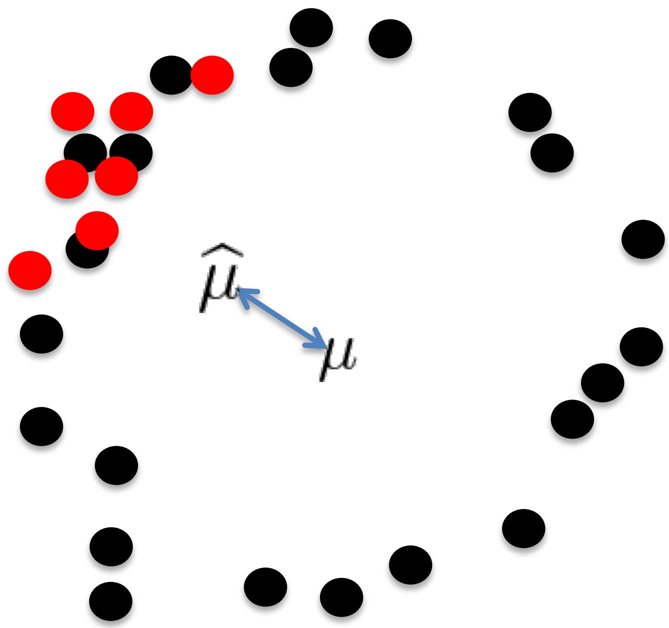


$$\widehat{\mu} \triangleq \frac{1}{N}\sum_{i=1}^{N} X_i$$

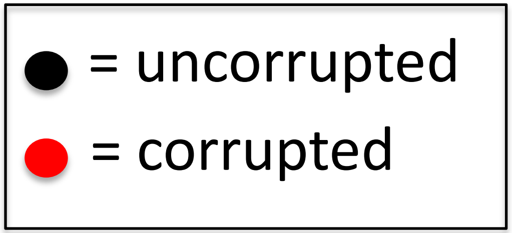


DETECTING CORRUPTIONS

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

(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

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

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- **A Win-Win Algorithm**
- Unknown Covariance

Part III: Experiments

A WIN-WIN ALGORITHM

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

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

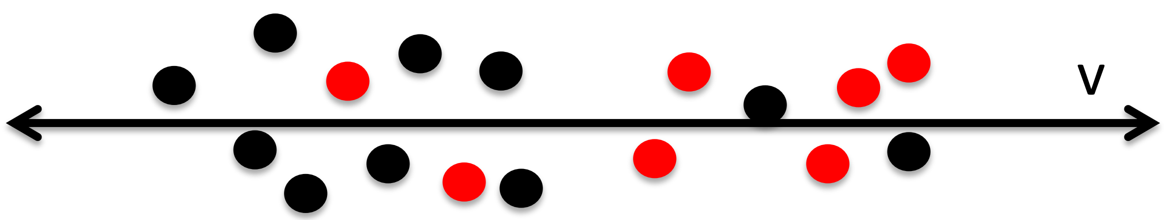
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points:



where v is the direction of largest variance

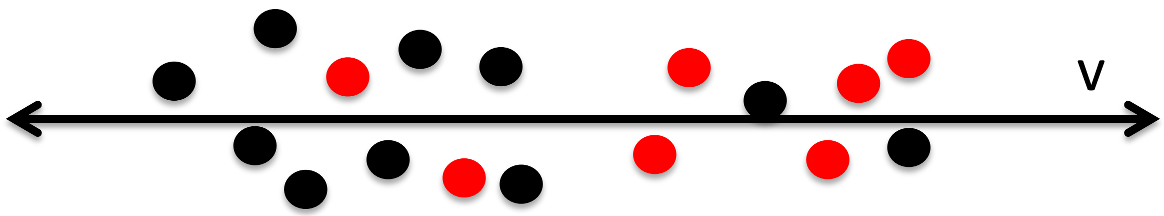
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points:



where v is the direction of largest variance, and T has a formula

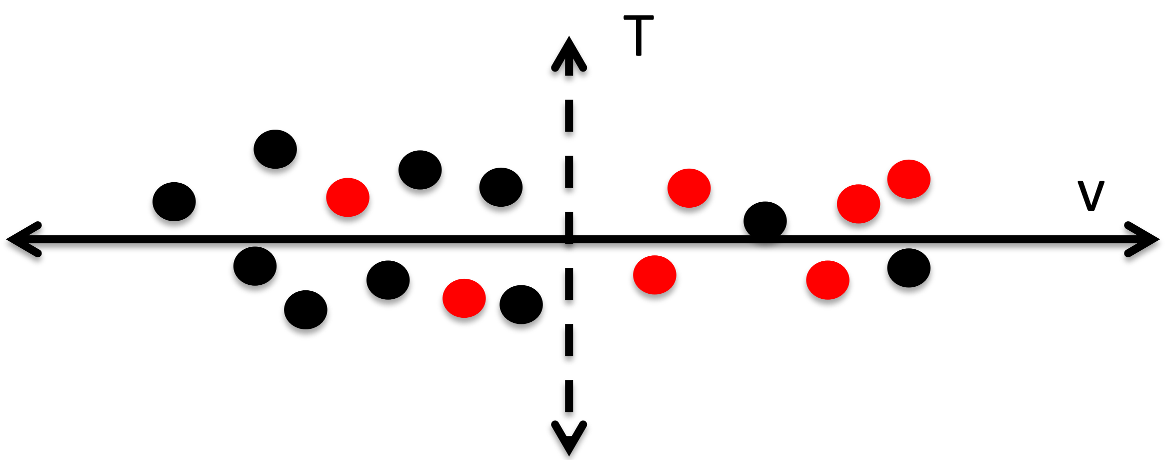
A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points:



where v is the direction of largest variance, and T has a formula

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points

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

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points

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

Eventually we find (certifiably) good parameters

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

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

A WIN-WIN ALGORITHM

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

Filtering Approach: Suppose that:

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

We can throw out more corrupted than uncorrupted points

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- A Win-Win Algorithm
- **Unknown Covariance**

Part III: Experiments

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

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

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

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

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

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

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

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
- A Win-Win Algorithm
- Unknown Covariance

Part III: Experiments

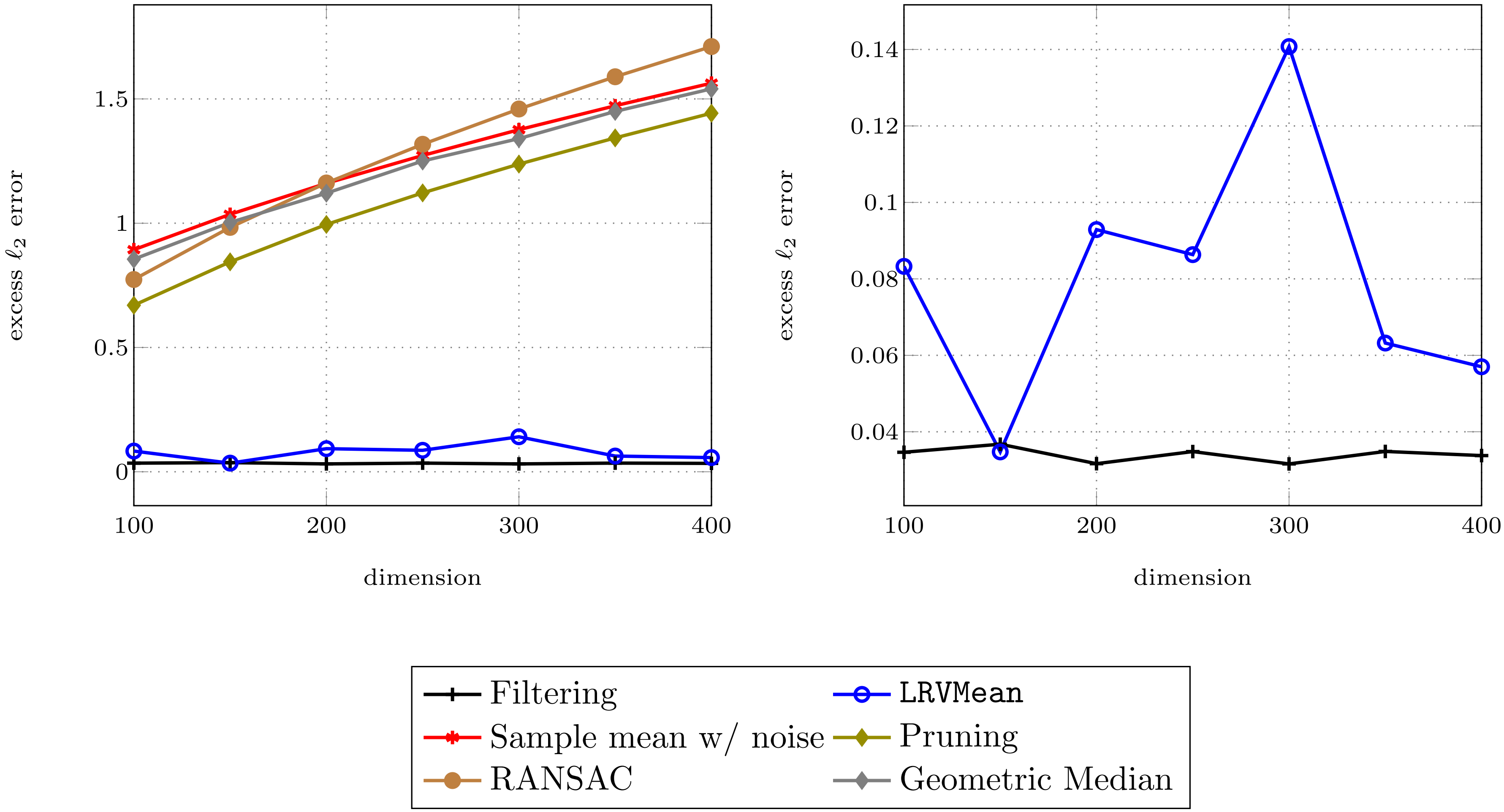
SYNTHETIC EXPERIMENTS

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

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

SYNTHETIC EXPERIMENTS

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



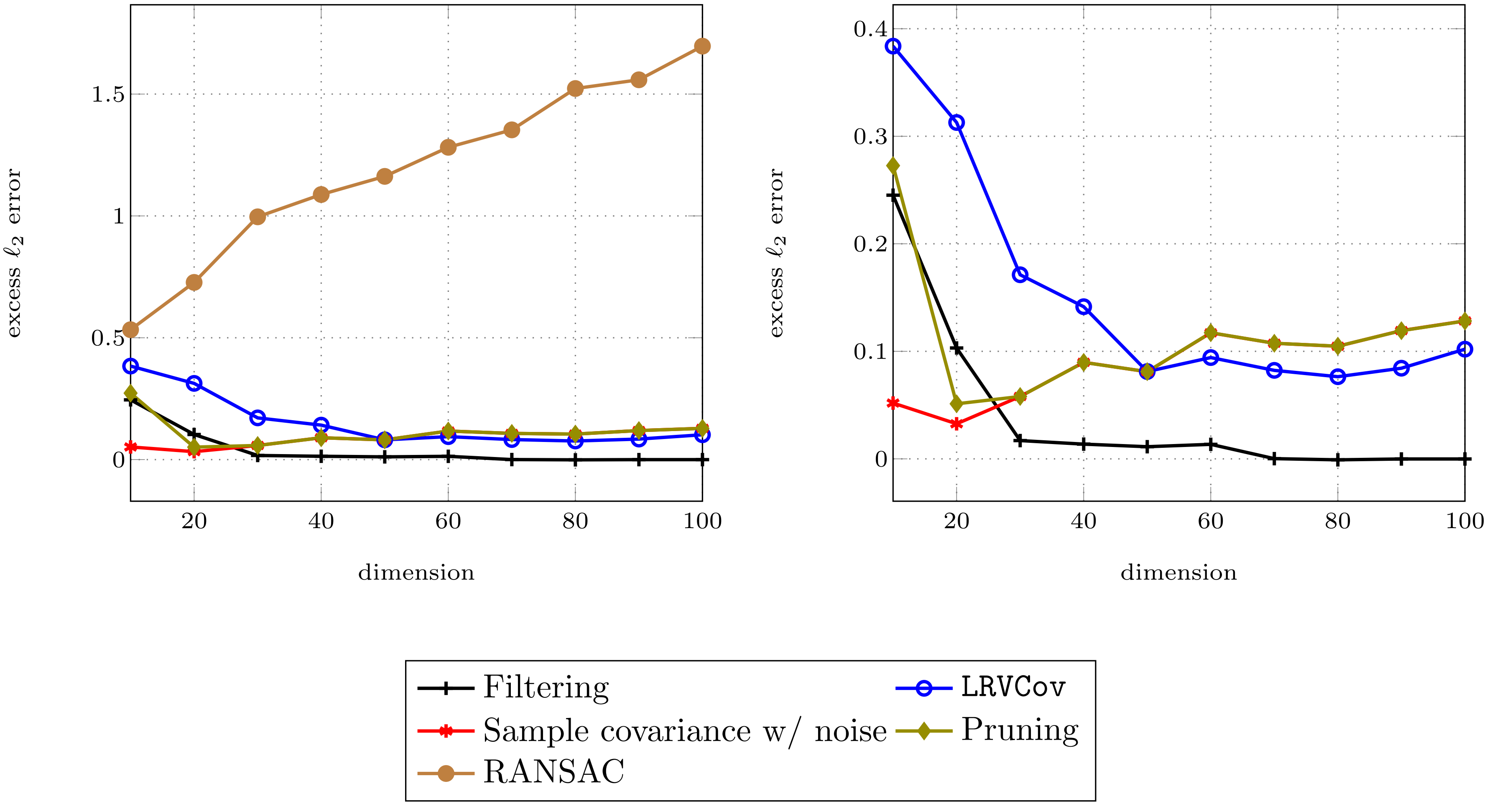
SYNTHETIC EXPERIMENTS

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

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

SYNTHETIC EXPERIMENTS

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



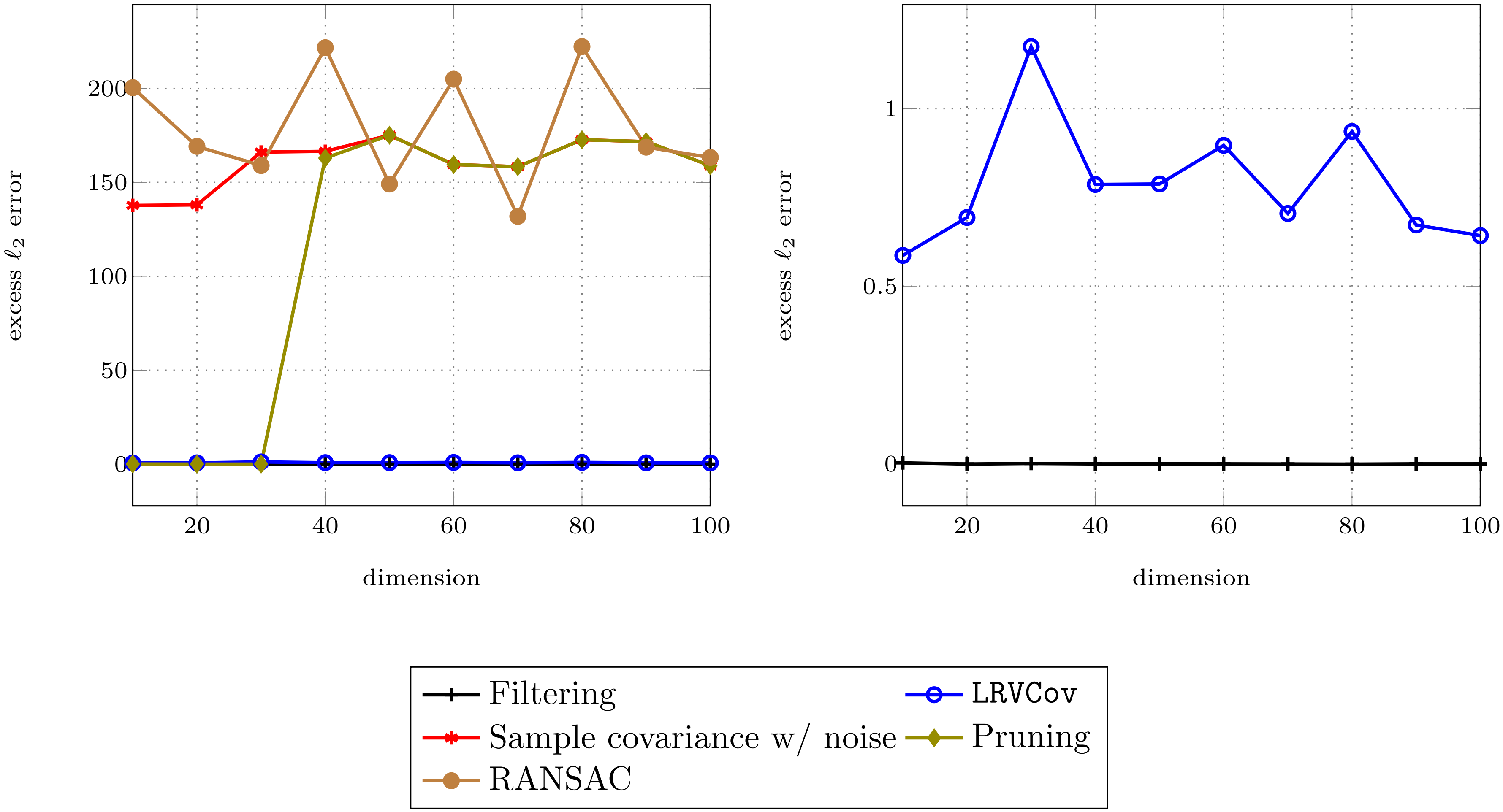
SYNTHETIC EXPERIMENTS

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

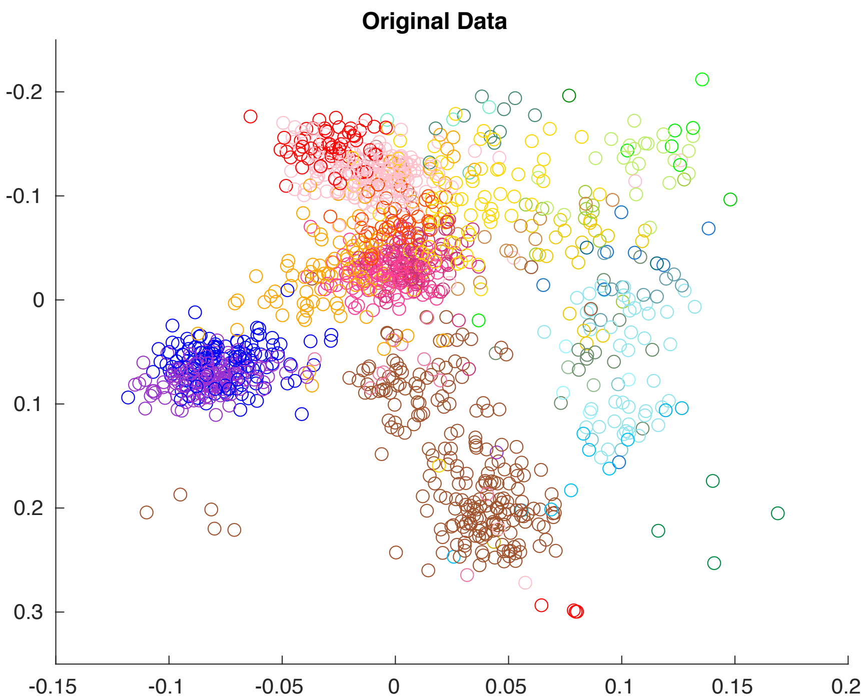
Famous study of **[Novembre et al. '08]**: Take top two singular vectors of people x SNP matrix (POPRES)

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"

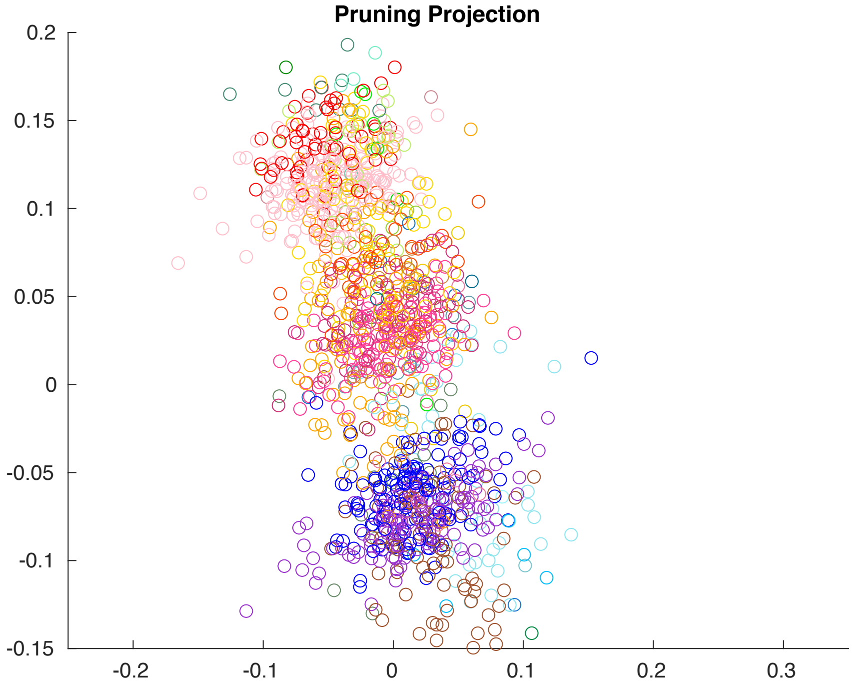
REAL DATA EXPERIMENTS

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

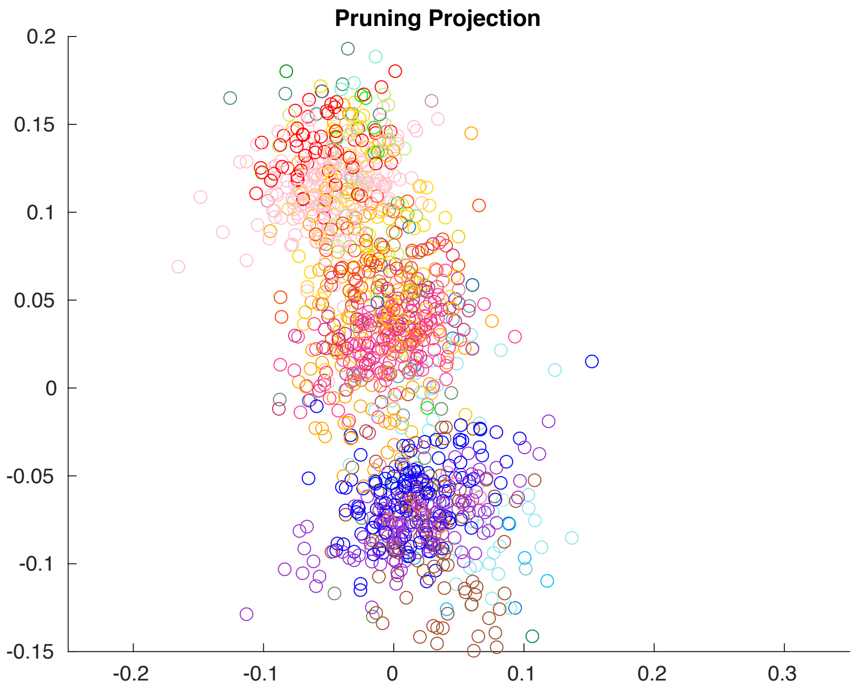


What PCA finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

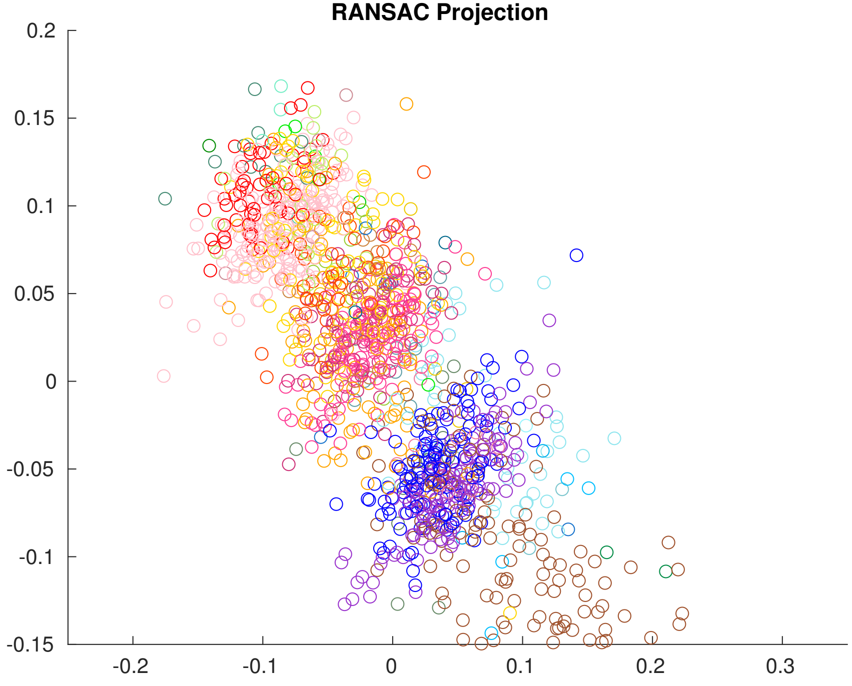


What PCA finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

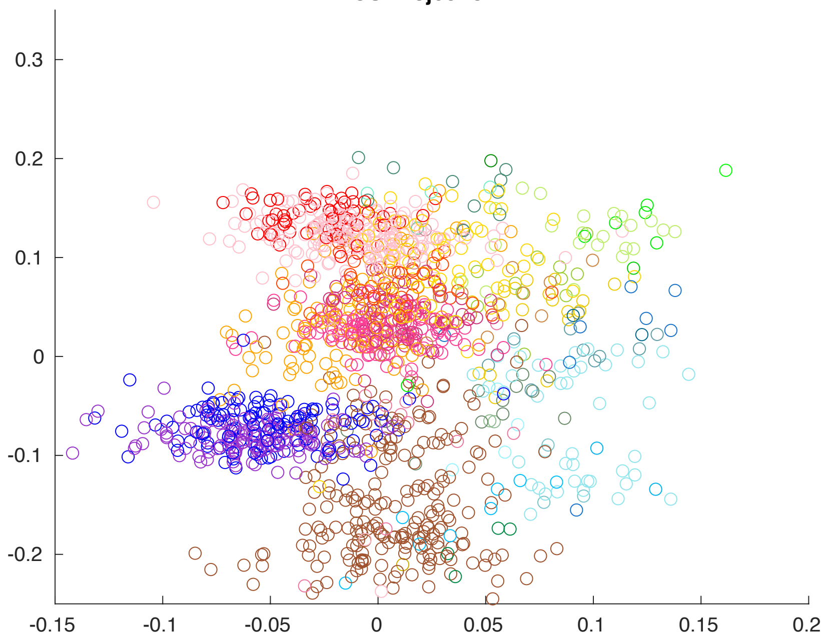


What RANSAC finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

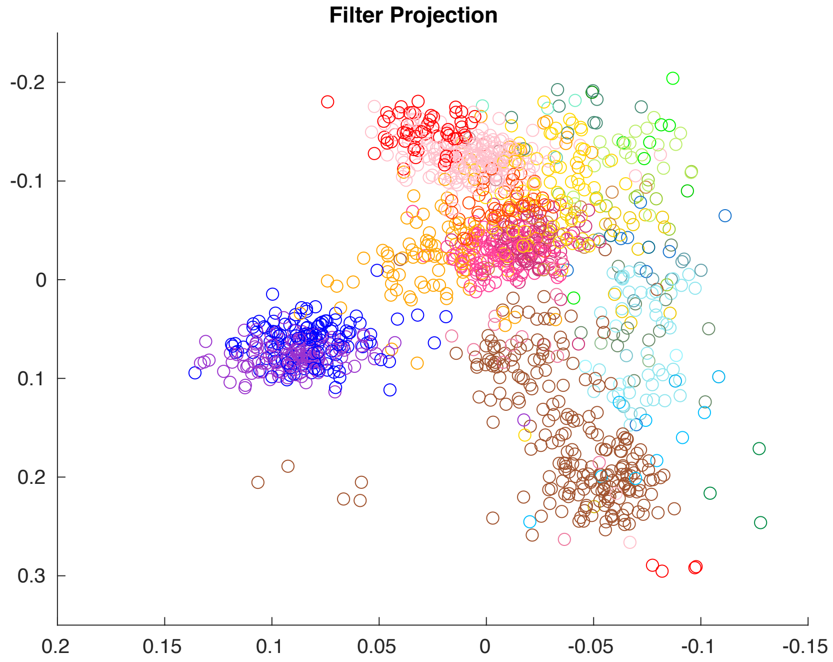


What robust PCA (via SDPs) finds

REAL DATA EXPERIMENTS

Can we find such patterns in the presence of noise?

10% noise

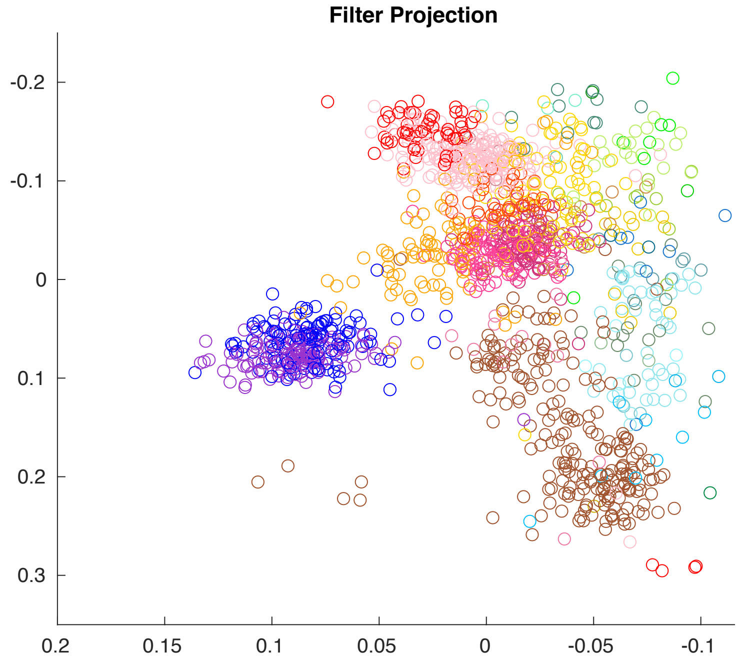


What our methods find

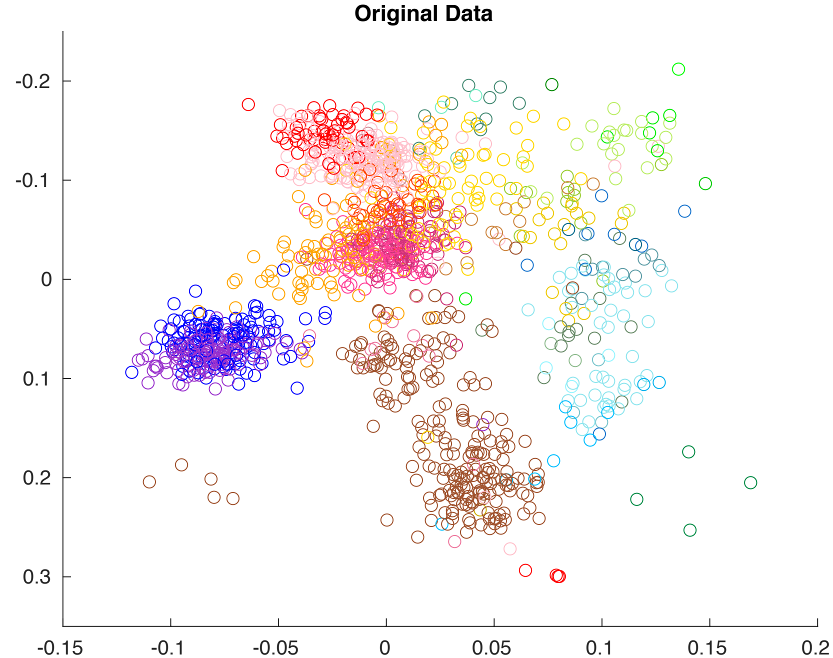
REAL DATA EXPERIMENTS

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?

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

OUTLINE

Part IV: Another Perspective on Robustness

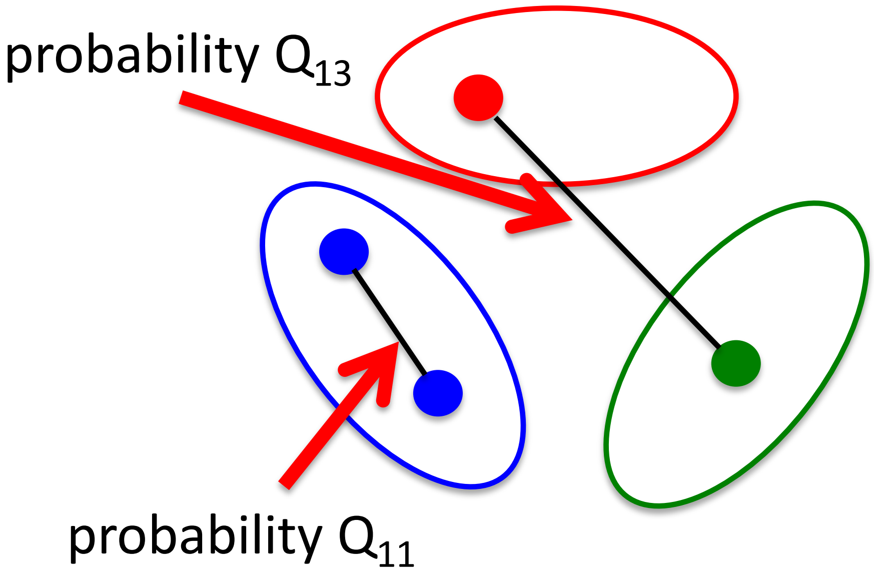
- **The Stochastic Block Model**
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

Let me tell you a story about the tension between **sharp thresholds** and **robustness**

THE STOCHASTIC BLOCK MODEL

Introduced by Holland, Laskey and Leinhardt (1983):



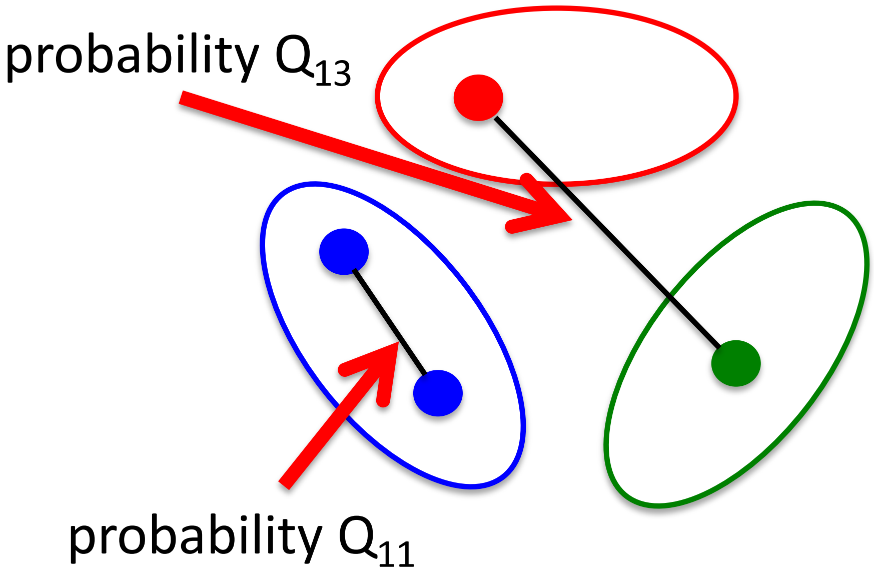
- k communities
- connection probabilities

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{32} \\ Q_{13} & Q_{32} & Q_{33} \end{bmatrix}$$

- edges independent

THE STOCHASTIC BLOCK MODEL

Introduced by Holland, Laskey and Leinhardt (1983):



- k communities
- connection probabilities

$$Q = \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{32} \\ Q_{13} & Q_{32} & Q_{33} \end{bmatrix}$$

- edges independent

Ubiquitous model studied in **statistics**, **computer science**, **information theory**, **statistical physics**

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting **[Bui, Chaudhuri, Leighton, Sipser ‘87]**

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting **[Bui, Chaudhuri, Leighton, Sipser ‘87]**

(2) Spectral Methods e.g. **[McSherry ‘01]**

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting **[Bui, Chaudhuri, Leighton, Sipser ‘87]**

(2) Spectral Methods e.g. **[McSherry ‘01]**

(3) Markov chain Monte Carlo (MCMC) e.g. **[Jerrum, Sorkin ‘98]**

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting **[Bui, Chaudhuri, Leighton, Sipser ‘87]**

(2) Spectral Methods e.g. **[McSherry ‘01]**

(3) Markov chain Monte Carlo (MCMC) e.g. **[Jerrum, Sorkin ‘98]**

(4) Semidefinite Programs e.g. **[Boppana ‘87]**

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting **[Bui, Chaudhuri, Leighton, Sipser '87]**

(2) Spectral Methods e.g. **[McSherry '01]**

(3) Markov chain Monte Carlo (MCMC) e.g. **[Jerrum, Sorkin '98]**

(4) Semidefinite Programs e.g. **[Boppana '87]**

These algorithms succeed in some ranges of parameters

Testbed for diverse range of algorithms

(1) Combinatorial Methods

e.g. degree counting [Bui, Chaudhuri, Leighton, Sipser '87]

(2) Spectral Methods e.g. [McSherry '01]

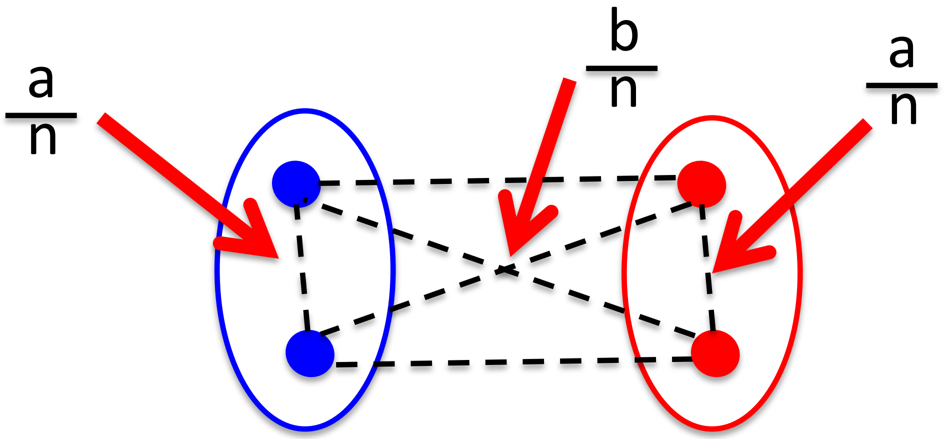
(3) Markov chain Monte Carlo (MCMC) e.g. [Jerrum, Sorkin '98]

(4) Semidefinite Programs e.g. [Boppana '87]

These algorithms succeed in some ranges of parameters

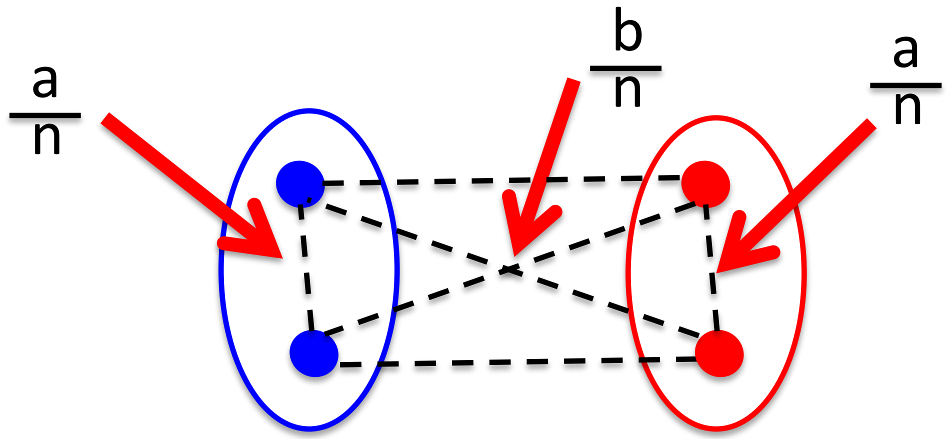
Can we reach the fundamental limits of the SBM?

Following Decelle, Krzakala, Moore and Zdeborová (2011), let's study the **sparse** regime:



where a, b = O(1) so that there are O(n) edges

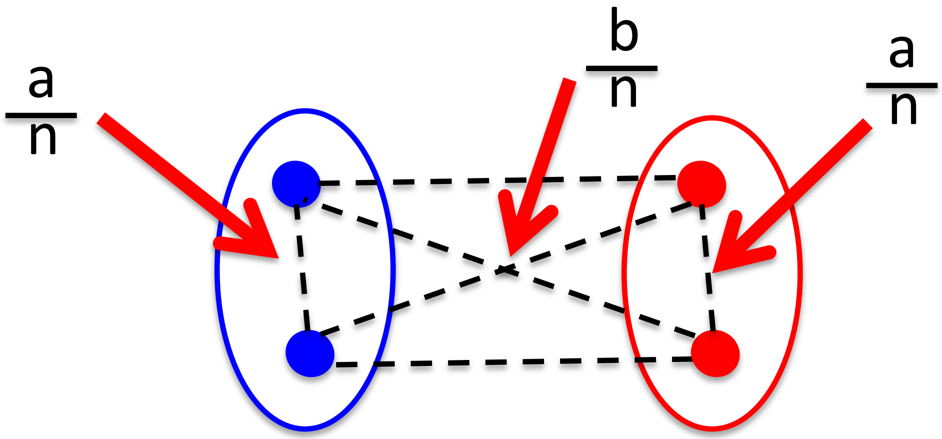
Following Decelle, Krzakala, Moore and Zdeborová (2011), let's study the **sparse** regime:



where a, b = O(1) so that there are O(n) edges

Remark: The degree of each node is Poi(a/2+b/2) hence there are many isolated nodes whose community we cannot find

Following Decelle, Krzakala, Moore and Zdeborová (2011), let's study the **sparse** regime:

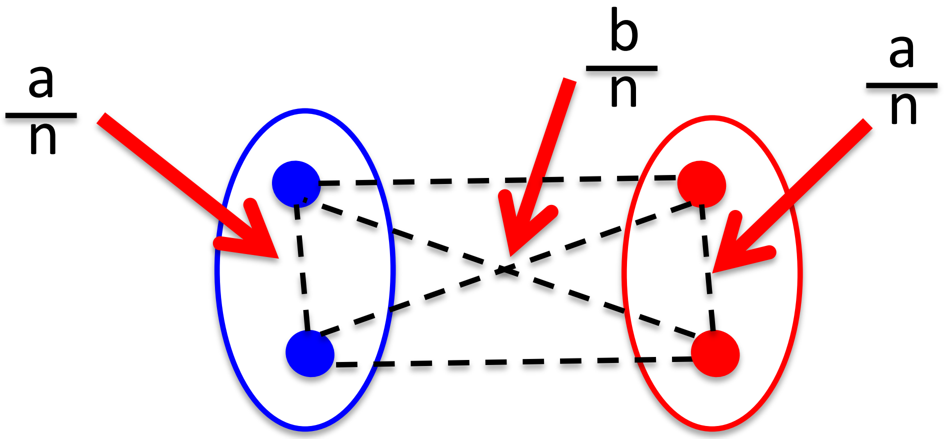


where a, b = O(1) so that there are O(n) edges

Remark: The degree of each node is Poi(a/2+b/2) hence there are many isolated nodes whose community we cannot find

Goal (Partial Recovery): Find a partition that has agreement better than ½ with true community structure

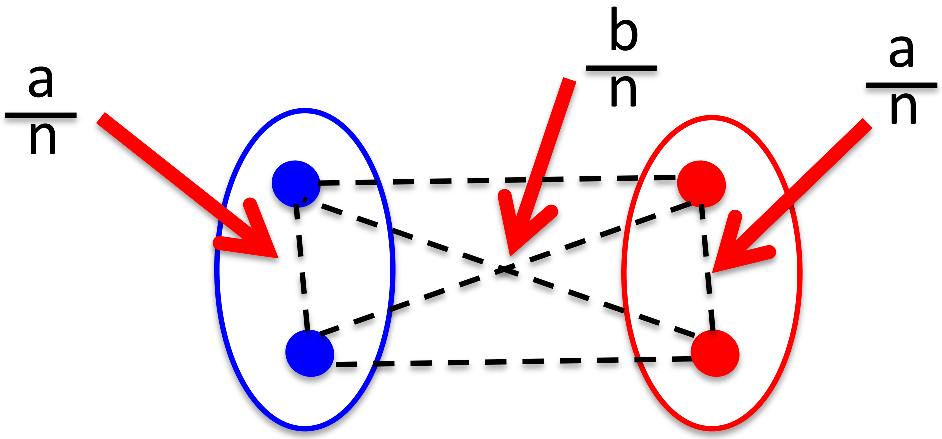
Following Decelle, Krzakala, Moore and Zdeborová (2011), let's study the **sparse** regime:



where a, b = O(1) so that there are O(n) edges

Conjecture: Partial recovery is possible iff $(a-b)^2 > 2(a+b)$

Following Decelle, Krzakala, Moore and Zdeborová (2011), let's study the **sparse** regime:



where a, b = O(1) so that there are O(n) edges

Conjecture: Partial recovery is possible iff $(a-b)^2 > 2(a+b)$

Conjecture is based on fixed points of **belief propagation**…

OUTLINE

Part IV: Another Perspective on. Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- **Belief Propagation and its Predictions**
- Semi-Random Models
- Sharpness vs. Robustness

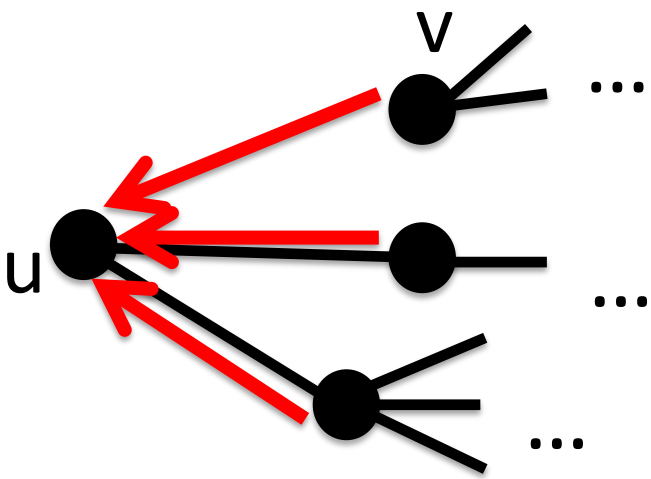
Part V: Above Average-Case?

BELIEF PROPAGATION

Introduced by Judea Pearl (1982):

"For fundamental contributions ... to probabilistic and causal reasoning"

Adapted to community detection:

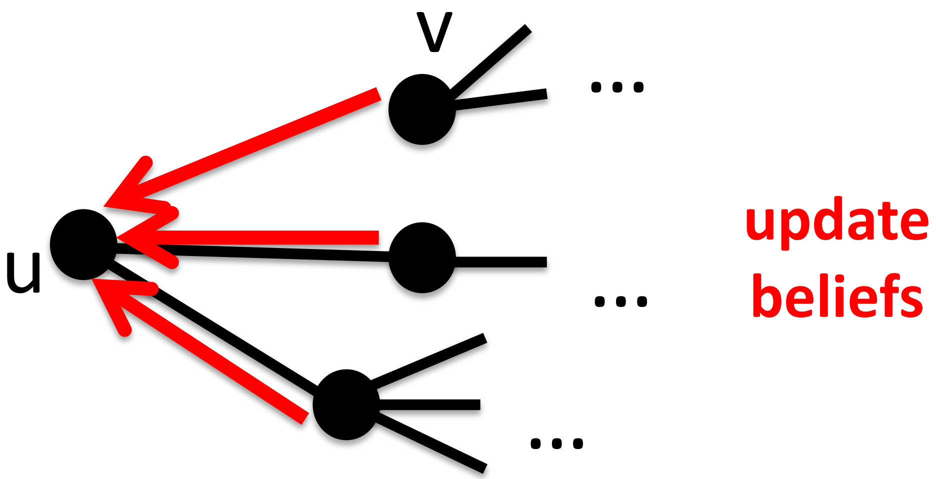


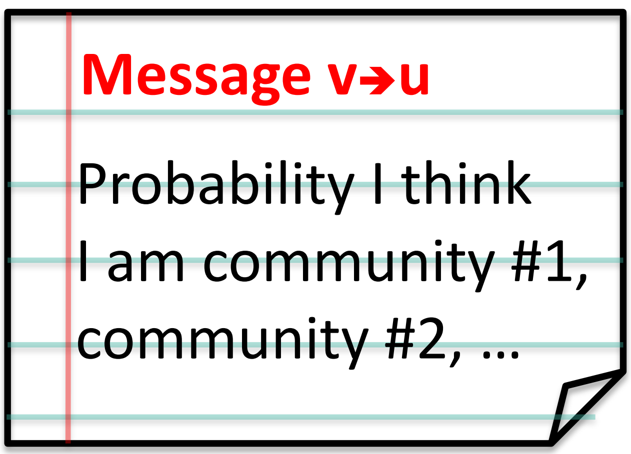
Message v→u

Probability I think
I am community #1,
community #2, ...

Do same for all nodes

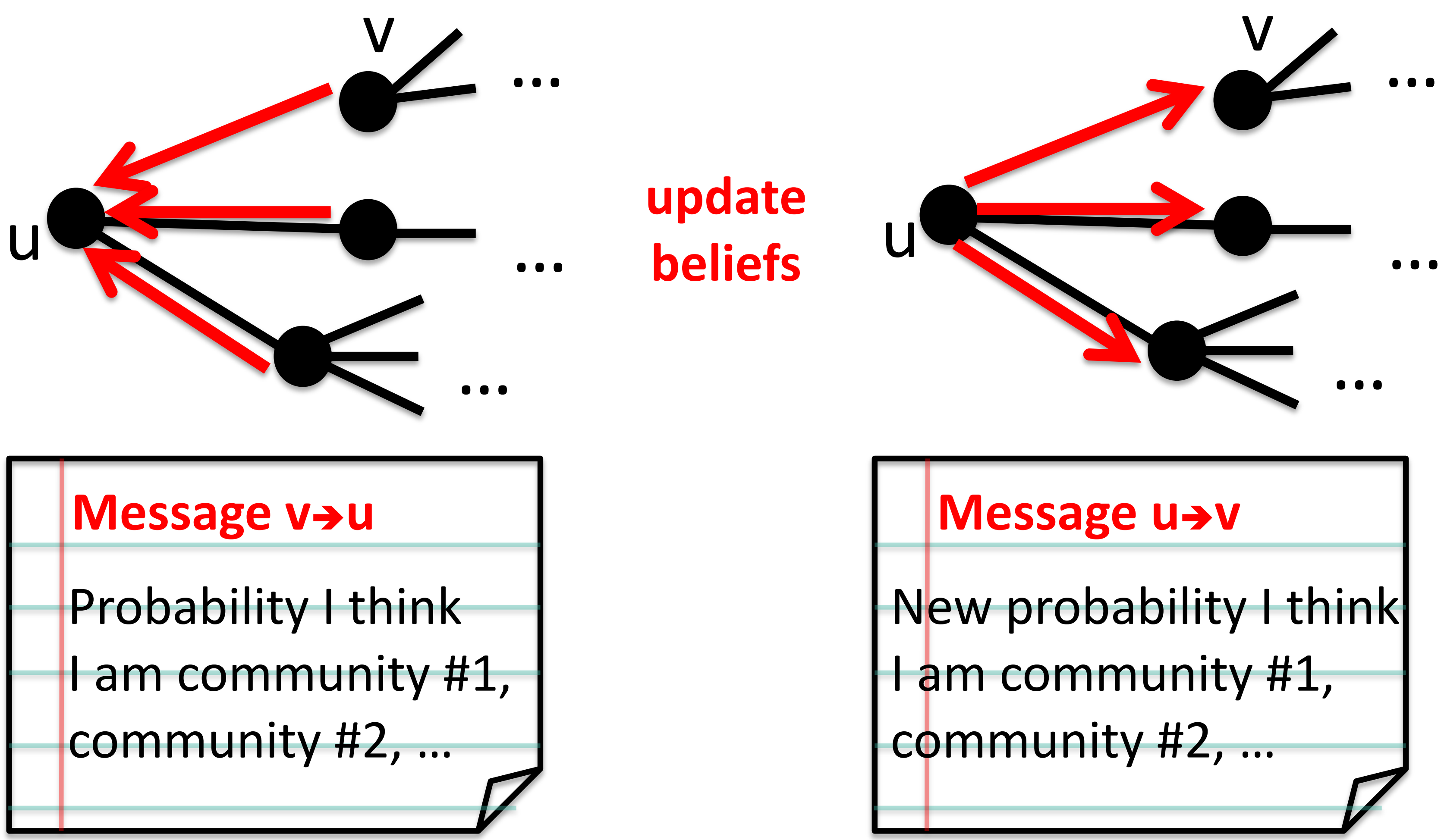
Adapted to community detection:





Do same for all nodes

Adapted to community detection:



Do same for all nodes

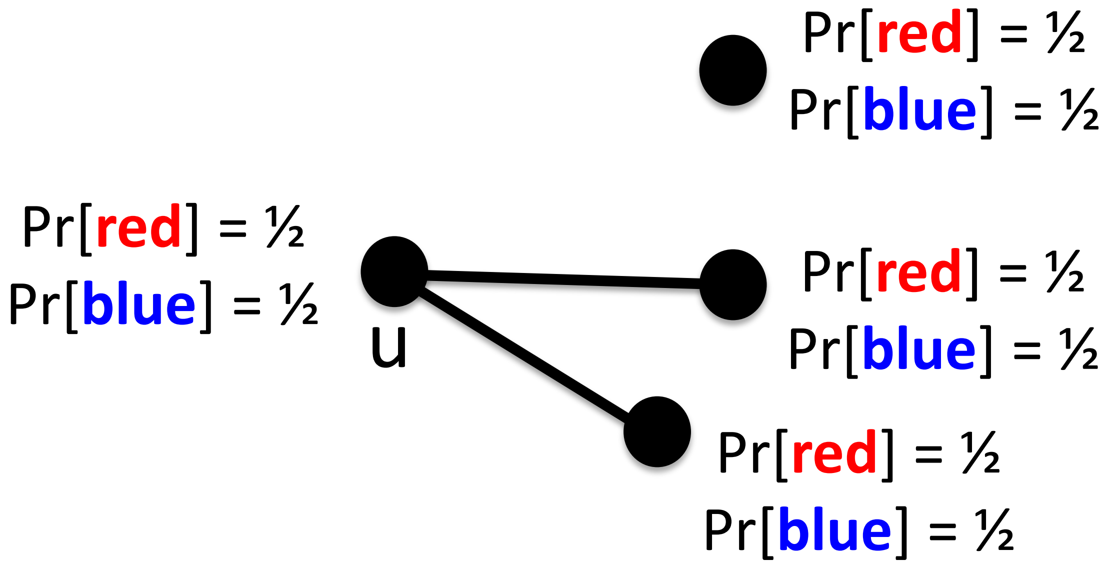
Do same for all nodes

THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck

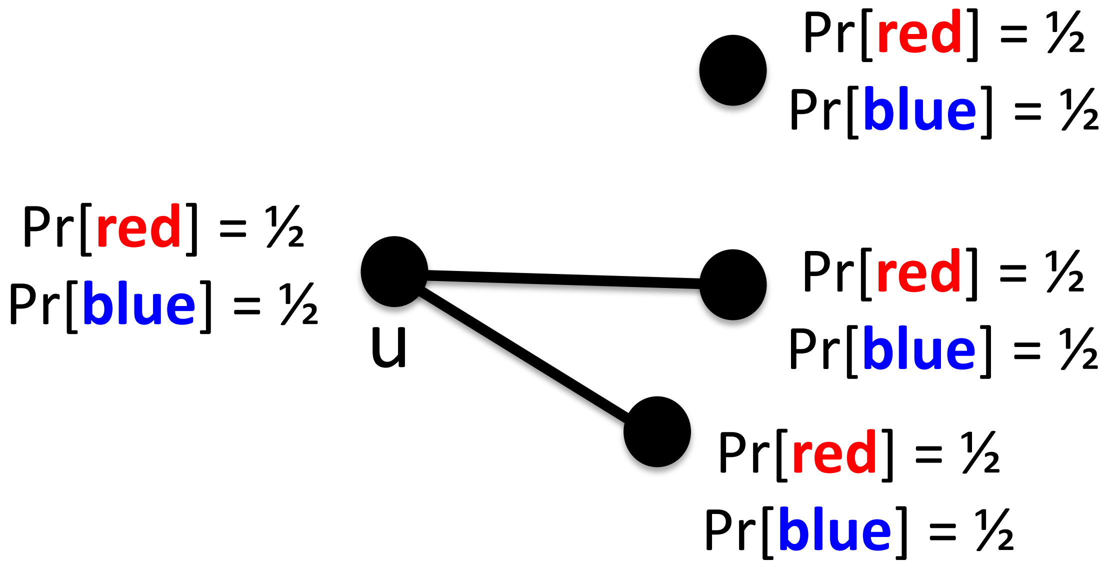
THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck



THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck



Claim: No one knows anything, **so you never have to update your beliefs**

THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck

Fact: If $(a-b)^2 > 2(a+b)$ then the trivial fixed point is unstable

THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck

Fact: If $(a-b)^2 > 2(a+b)$ then the trivial fixed point is unstable

Hope: Whatever it finds, solves partial recovery

THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck

Fact: If $(a-b)^2 > 2(a+b)$ then the trivial fixed point is unstable

Hope: Whatever it finds, solves partial recovery

Evidence based on simulations

THE TRIVIAL FIXED POINT

Belief propagation has a trivial fixed point where it gets stuck

Fact: If $(a-b)^2 > 2(a+b)$ then the trivial fixed point is unstable

Hope: Whatever it finds, solves partial recovery

Evidence based on simulations

And if $(a-b)^2 \le 2(a+b)$ and it does get stuck, then maybe partial recovery is **information theoretically impossible**?

CONJECTURE IS PROVED!

Mossel, Neeman and Sly (2013) and Massoulie (2013):

Theorem: It is possible to find a partition that is correlated with true communities iff $(a-b)^2 > 2(a+b)$

CONJECTURE IS PROVED!

Mossel, Neeman and Sly (2013) and Massoulie (2013):

Theorem: It is possible to find a partition that is correlated with true communities iff $(a-b)^2 > 2(a+b)$

Later attempts based on SDPs only get to

$$(a-b)^2 > C(a+b), \text{ for some } C > 2$$

CONJECTURE IS PROVED!

Mossel, Neeman and Sly (2013) and Massoulie (2013):

Theorem: It is possible to find a partition that is correlated with true communities iff $(a-b)^2 > 2(a+b)$

Later attempts based on SDPs only get to

$$(a-b)^2 > C(a+b), \text{ for some } C > 2$$

Are nonconvex methods **better** than convex programs?

CONJECTURE IS PROVED!

Mossel, Neeman and Sly (2013) and Massoulie (2013):

Theorem: It is possible to find a partition that is correlated with true communities iff $(a-b)^2 > 2(a+b)$

Later attempts based on SDPs only get to

$$(a-b)^2 > C(a+b), \text{ for some } C > 2$$

Are nonconvex methods **better** than convex programs?

How do predictions of statistical physics and SDPs compare?

CONJECTURE IS PROVED!

Mossel, Neeman and Sly (2013) and Massoulie (2013):

Theorem: It is possible to find a partition that is correlated with true communities iff $(a\text{-}b)^2 > 2(a+b)$

Later attempts based on SDPs only get to

$$(a\text{-}b)^2 > C(a+b), \text{ for some } C > 2$$

Are nonconvex methods **better** than convex programs?

How do predictions of statistical physics and SDPs compare?

Robustness will be a key player in the answers

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- **Semi-Random Models**
- Sharpness vs. Robustness

Part V: Above Average-Case?

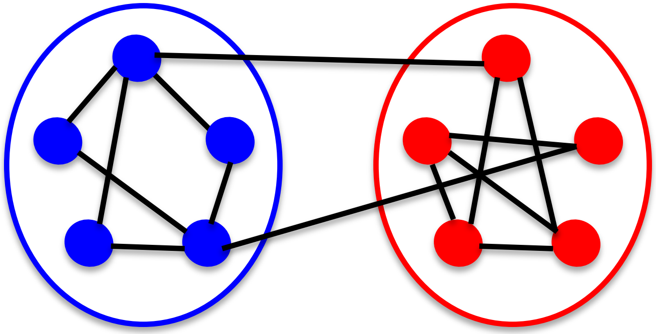
SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

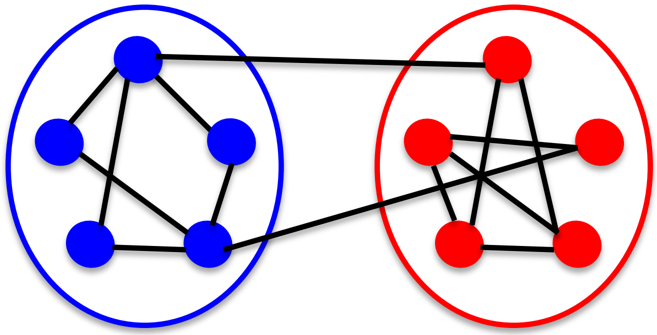
(1) Sample graph from SBM



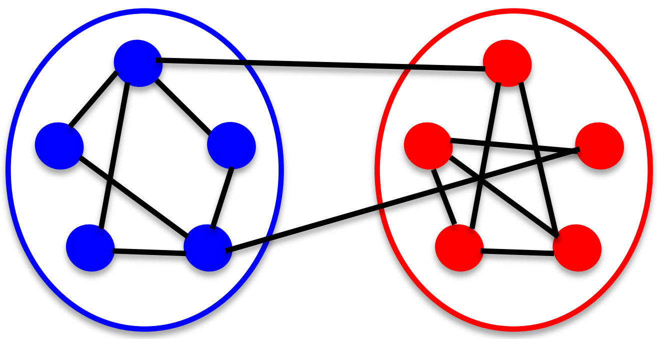
SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

(1) Sample graph from SBM



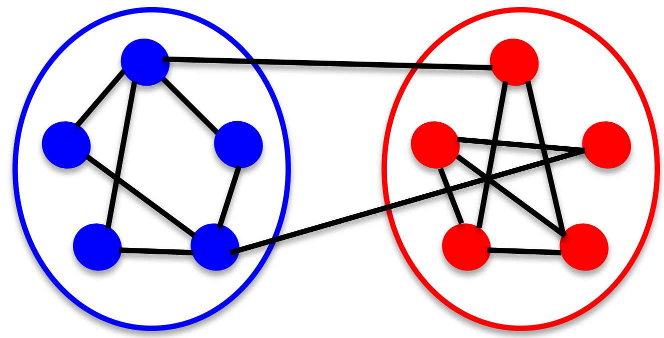
(2) Adversary can add edges within community and delete edges crossing



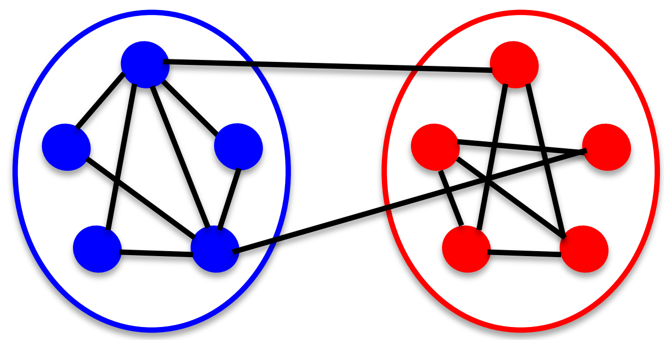
SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

(1) Sample graph from SBM



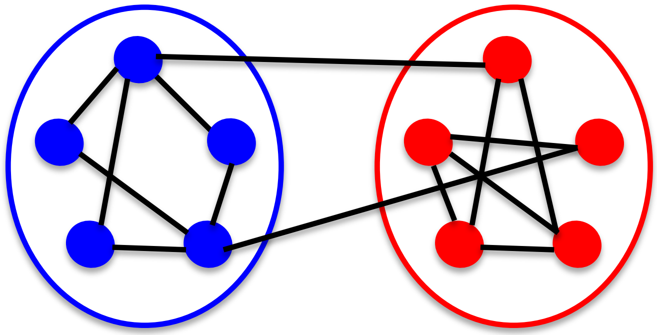
(2) Adversary can add edges within community and delete edges crossing



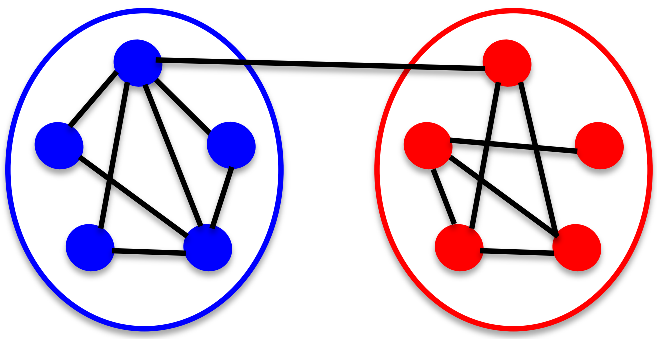
SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

(1) Sample graph from SBM



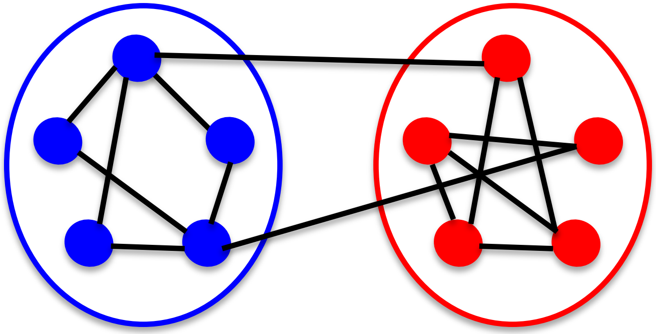
(2) Adversary can add edges
within community
and delete edges crossing



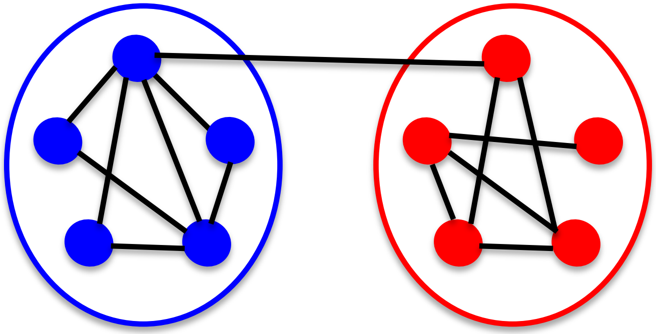
SEMI-RANDOM MODELS

Introduced by Blum and Spencer (1995), Feige and Kilian (2001):

(1) Sample graph from SBM



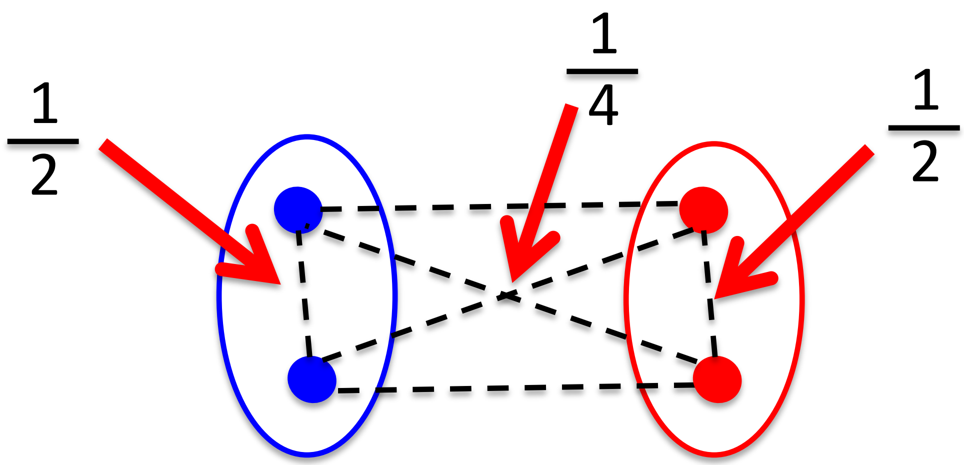
(2) Adversary can add edges within community and delete edges crossing



Algorithms can no longer over tune to distribution

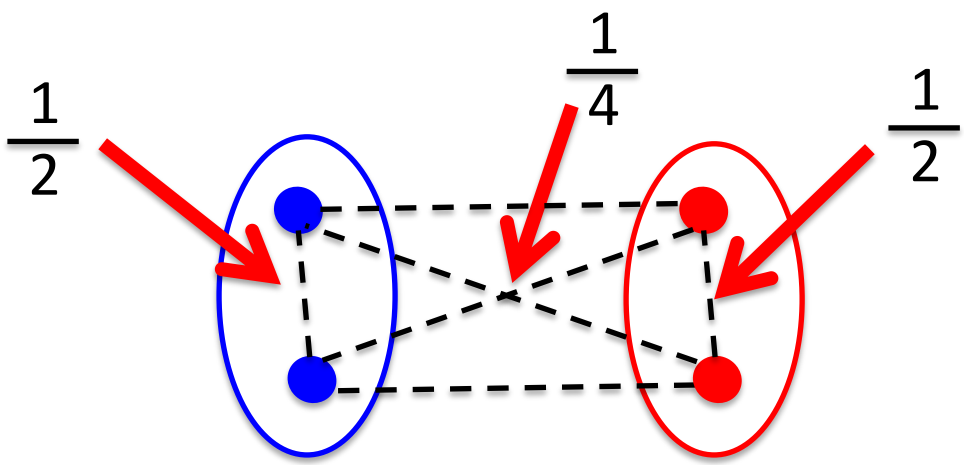
A NON-ROBUST ALGORITHM

Consider the following SBM:



A NON-ROBUST ALGORITHM

Consider the following SBM:

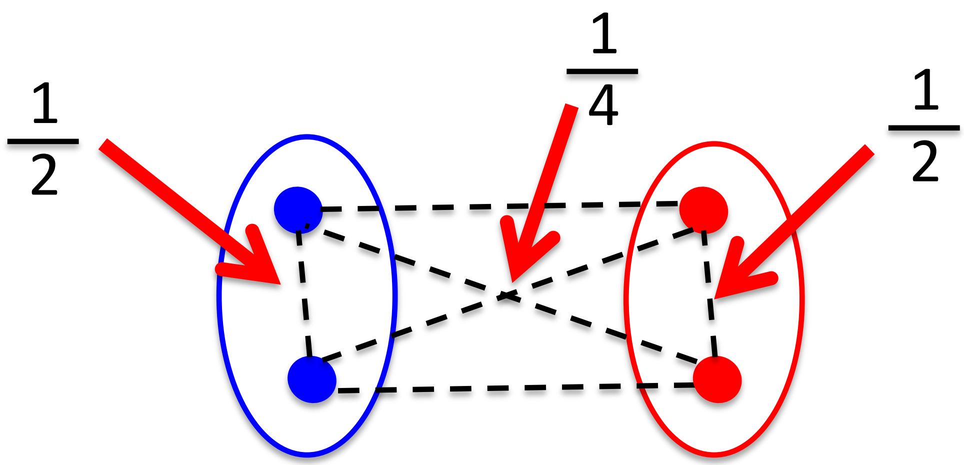


Number of common neighbors

Nodes from same community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

A NON-ROBUST ALGORITHM

Consider the following SBM:



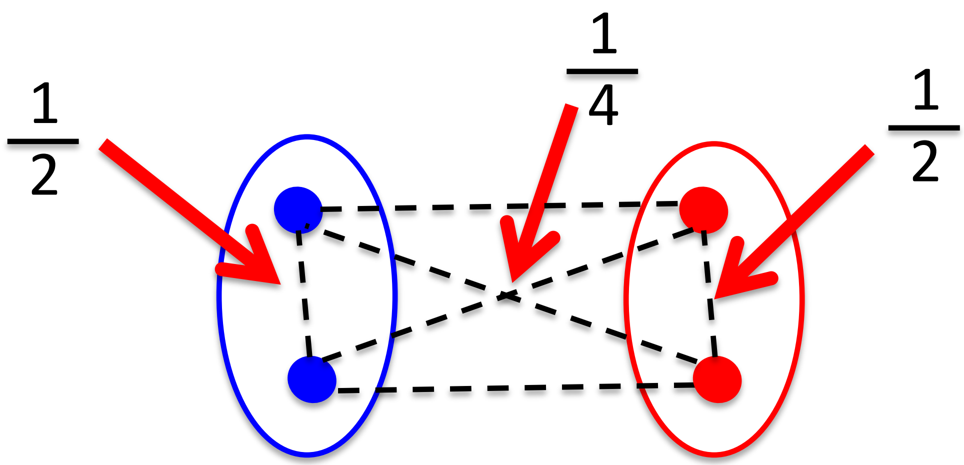
Number of common neighbors

Nodes from same community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

Nodes from diff. community: $\left(\frac{1}{2}\right)\left(\frac{1}{4}\right) n$

A NON-ROBUST ALGORITHM

Consider the following SBM:



Number of common neighbors

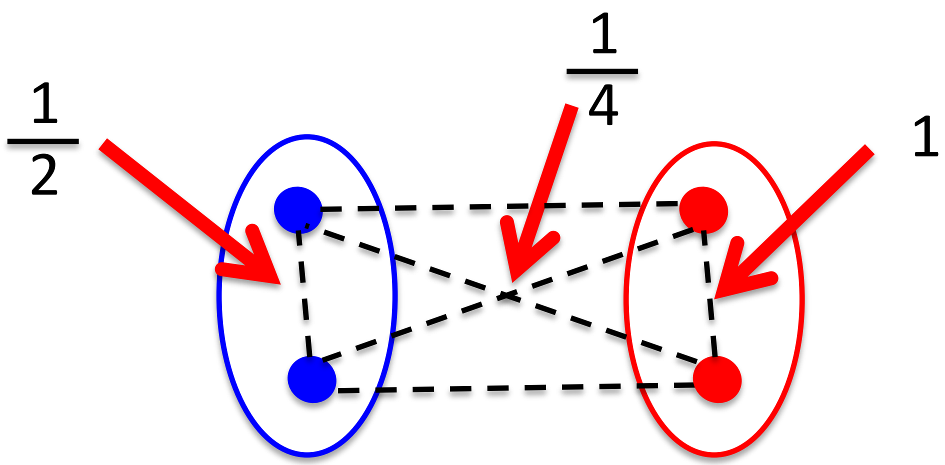
Nodes from same community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

$\vee$

Nodes from diff. community: $\left(\frac{1}{2}\right)\left(\frac{1}{4}\right) n$

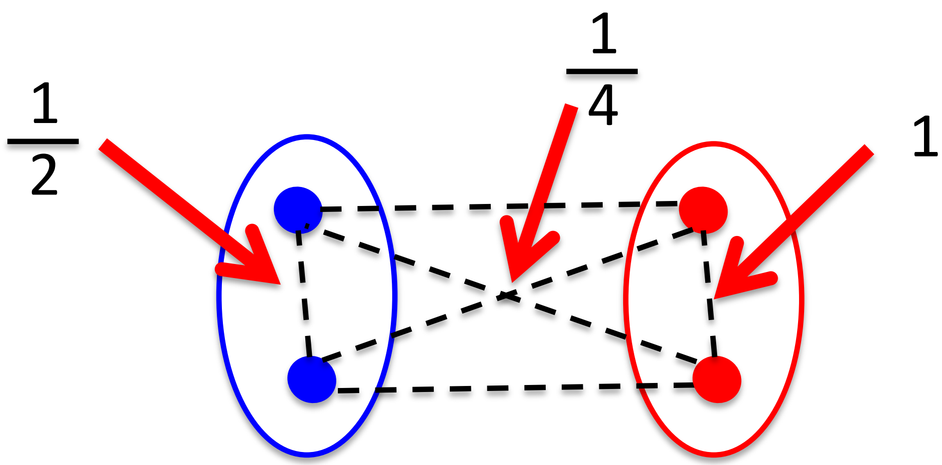
A NON-ROBUST ALGORITHM

Semi-random adversary: Add clique to **red** community



A NON-ROBUST ALGORITHM

Semi-random adversary: Add clique to **red** community

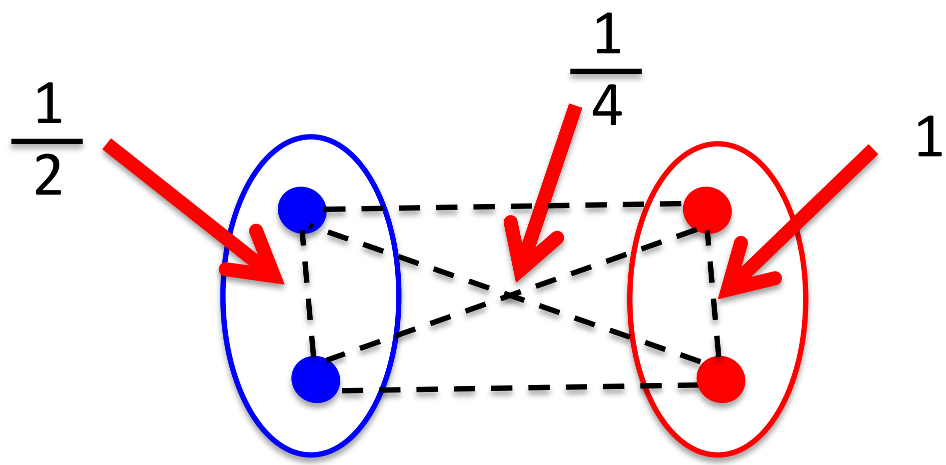


Number of common neighbors

Nodes from **blue** community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

A NON-ROBUST ALGORITHM

Semi-random adversary: Add clique to **red** community



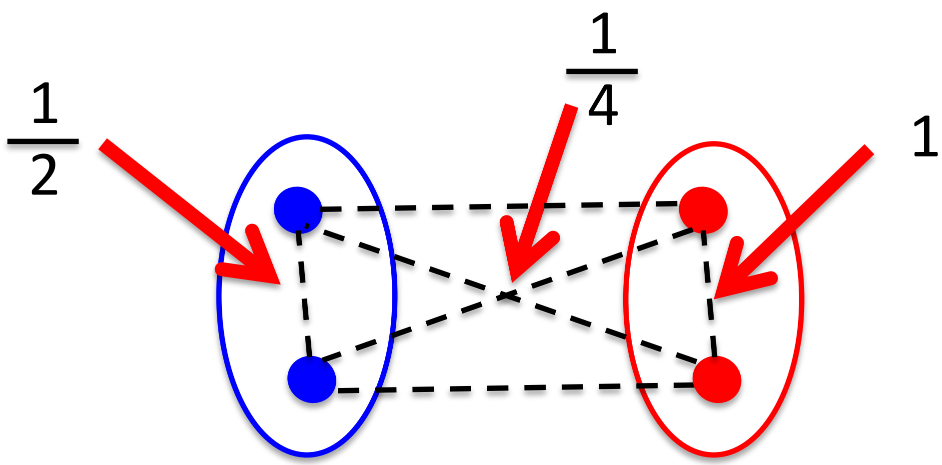
Number of common neighbors

Nodes from **blue** community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

Nodes from diff. community: $\left(\frac{1}{2}\right)\left(\frac{1}{4}\right)\frac{n}{2} + \left(\frac{1}{4}\right)\frac{n}{2}$

A NON-ROBUST ALGORITHM

Semi-random adversary: Add clique to **red** community



Number of common neighbors

Nodes from **blue** community: $\left(\frac{1}{2}\right)^2 \frac{n}{2} + \left(\frac{1}{4}\right)^2 \frac{n}{2}$

$\wedge$

Nodes from diff. community: $\left(\frac{1}{2}\right)\left(\frac{1}{4}\right)\frac{n}{2} + \left(\frac{1}{4}\right)\frac{n}{2}$

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- **Sharpness vs. Robustness**

Part V: Above Average-Case?

SHARPNESS VS. ROBUSTNESS

Monotone changes break most algorithms, in fact something more fundamental is happening:

SHARPNESS VS. ROBUSTNESS

Monotone changes break most algorithms, in fact something more fundamental is happening:

Theorem [Moitra, Perry, Wein '16]: It is **information theoretically impossible** to recover a partition correlated with true communities for $(a-b)^2 \leq C_{a,b}(a+b)$ for some $C_{a,b} > 2$ in the semirandom model

SHARPNESS VS. ROBUSTNESS

Monotone changes break most algorithms, in fact something more fundamental is happening:

Theorem [Moitra, Perry, Wein '16]: It is **information theoretically impossible** to recover a partition correlated with true communities for $(a-b)^2 \leq C_{a,b}(a+b)$ for some $C_{a,b} > 2$ in the semirandom model

But SDPs continue to work in semirandom model

SHARPNESS VS. ROBUSTNESS

Monotone changes break most algorithms, in fact something more fundamental is happening:

Theorem [Moitra, Perry, Wein '16]: It is **information theoretically impossible** to recover a partition correlated with true communities for $(a-b)^2 \leq C_{a,b}(a+b)$ for some $C_{a,b} > 2$ in the semirandom model

But SDPs continue to work in semirandom model

Being robust can make the problem strictly harder, but why?

SHARPNESS VS. ROBUSTNESS

Monotone changes break most algorithms, in fact something more fundamental is happening:

Theorem [Moitra, Perry, Wein '16]: It is **information theoretically impossible** to recover a partition correlated with true communities for $(a-b)^2 \leq C_{a,b}(a+b)$ for some $C_{a,b} > 2$ in the semirandom model

But SDPs continue to work in semirandom model

Being robust can make the problem strictly harder, but why?

Reaching the sharp threshold for community detection requires exploiting the structure of the noise

OUTLINE

Part IV: Another Perspective on Robustness

- The Stochastic Block Model
- Belief Propagation and its Predictions
- Semi-Random Models
- Sharpness vs. Robustness

Part V: Above Average-Case?

Models are a measuring stick to compare algorithms, but are we studying the right ones?

Models are a measuring stick to compare algorithms, but are we studying the right ones?

Average-case models: When we have many algorithms, can we find the *best* one?

Models are a measuring stick to compare algorithms, but are we studying the right ones?

Average-case models: When we have many algorithms, can we find the *best* one?

Semi-random models: When SDPs work, they're not exploiting the structure of the noise

BETWEEN WORST-CASE AND AVERAGE-CASE

Spielman and Teng (2001):

> "Explain why algorithms work well in practice, despite bad worst-case behavior"

Usually called *Beyond Worst-Case Analysis*

BETWEEN WORST-CASE AND AVERAGE-CASE

Spielman and Teng (2001):

"Explain why algorithms work well in practice, despite bad worst-case behavior"

Usually called *Beyond Worst-Case Analysis*

Semirandom models as *Above Average-Case Analysis*?

BETWEEN WORST-CASE AND AVERAGE-CASE

Spielman and Teng (2001):

"Explain why algorithms work well in practice, despite bad worst-case behavior"

Usually called *Beyond Worst-Case Analysis*

Semirandom models as *Above Average-Case Analysis*?

What else are we missing, if we only study problems in the average-case?

LOOKING FORWARD

Are there nonconvex methods that match the robustness guarantees of convex relaxations?

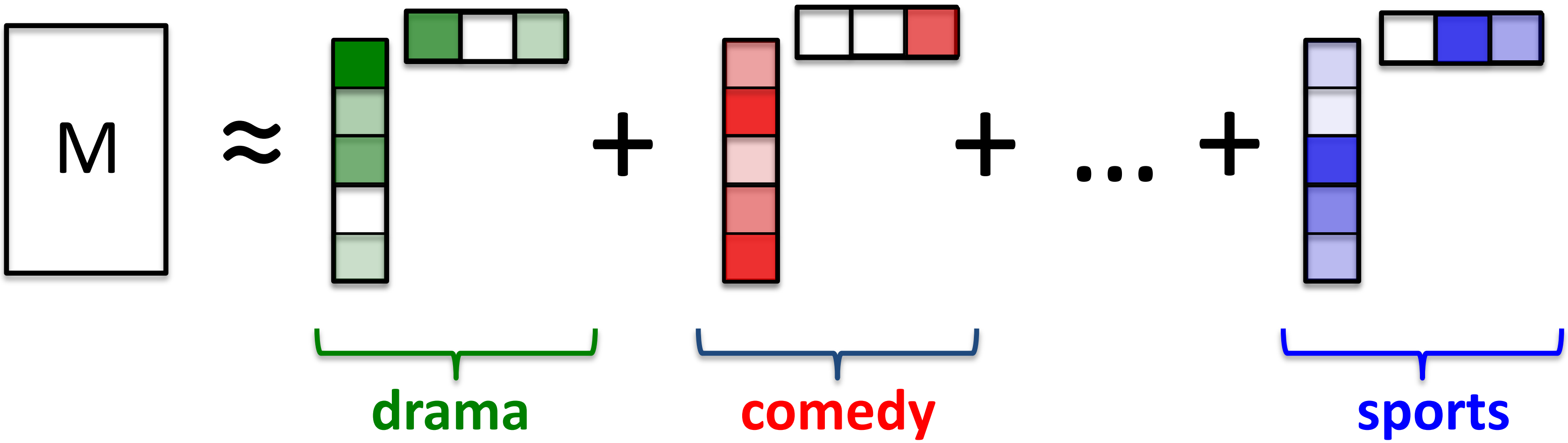
LOOKING FORWARD

Are there nonconvex methods that match the robustness guarantees of convex relaxations?

What models of robustness make sense for your favorite problems?

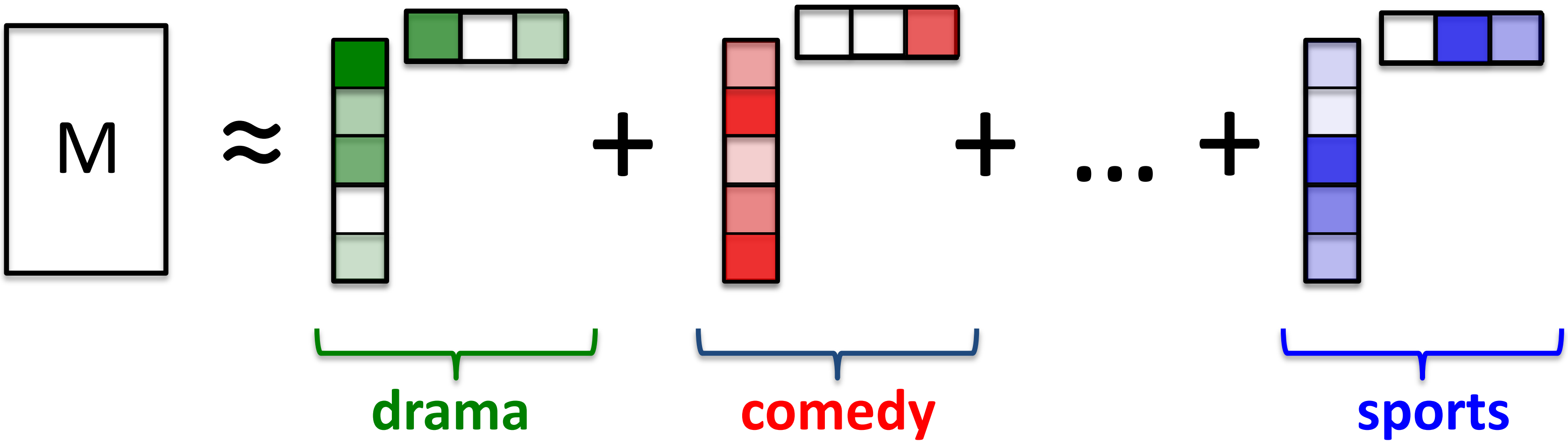
THE NETFLIX PROBLEM

Let M be an unknown, low-rank matrix



THE NETFLIX PROBLEM

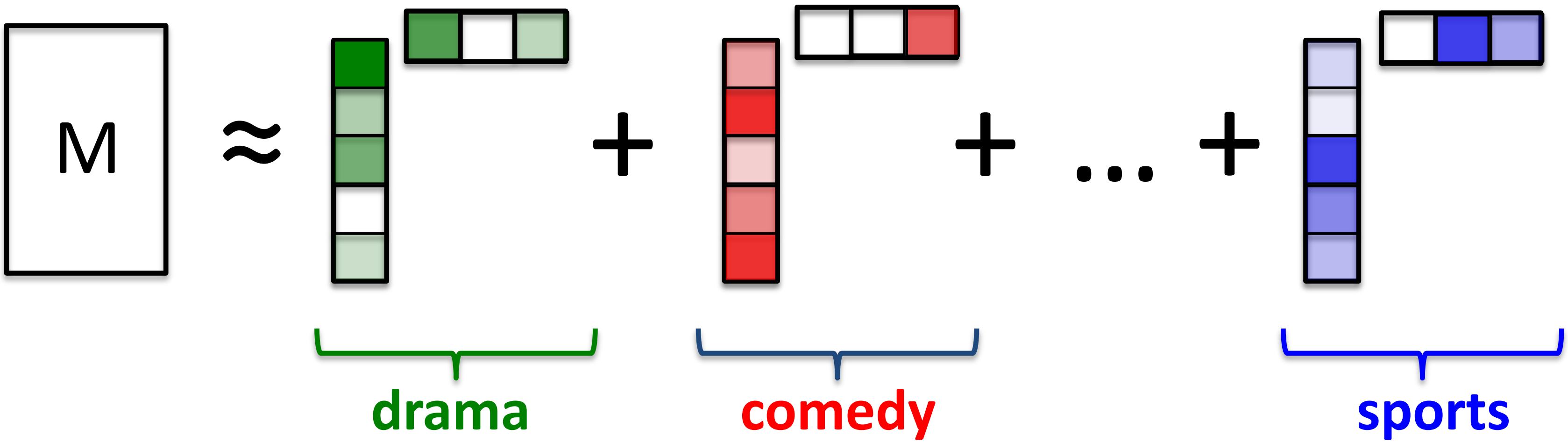
Let M be an unknown, low-rank matrix



Model: We are given random observations $M_{i,j}$ for all $i,j \in \Omega$

THE NETFLIX PROBLEM

Let M be an unknown, low-rank matrix



Model: We are given random observations $M_{i,j}$ for all $i,j \in \Omega$

Is there an efficient algorithm to recover M?

CONVEX PROGRAMMING APPROACH

$$\min \|X\|_* \text{ s.t. } \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \quad \textbf{(P)}$$

Here $\|X\|_*$ is the **nuclear norm**, i.e. sum of the singular values of X

[Fazel], [Srebro, Shraibman], [Recht, Fazel, Parrilo], [Candes, Recht], [Candes, Tao], [Candes, Plan], [Recht],

CONVEX PROGRAMMING APPROACH

$$\min \|X\|_* \text{ s.t. } \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \quad \textbf{(P)}$$

Here $\|X\|_*$ is the **nuclear norm**, i.e. sum of the singular values of X

[Fazel], [Srebro, Shraibman], [Recht, Fazel, Parrilo], [Candes, Recht], [Candes, Tao], [Candes, Plan], [Recht],

Theorem: If M is n x n and has rank r, and is C-incoherent then **(P)** recovers M exactly from $C^6 nr\log^2 n$ observations

ALTERNATING MINIMIZATION

Repeat:

$$U \leftarrow \underset{U}{\operatorname{argmin}} \sum_{(i,j)\in\Omega} |(UV^T)_{i,j} - M_{i,j}|^2$$

$$V \leftarrow \underset{V}{\operatorname{argmin}} \sum_{(i,j)\in\Omega} |(UV^T)_{i,j} - M_{i,j}|^2$$

[Keshavan, Montanari, Oh], [Jain, Netrapalli, Sanghavi], [Hardt]

ALTERNATING MINIMIZATION

Repeat:
$$U \leftarrow \underset{U}{\operatorname{argmin}} \sum_{(i,j)\in\Omega} \left|(UV^T)_{i,j} - M_{i,j}\right|^2$$

$$V \leftarrow \underset{V}{\operatorname{argmin}} \sum_{(i,j)\in\Omega} \left|(UV^T)_{i,j} - M_{i,j}\right|^2$$

[Keshavan, Montanari, Oh], [Jain, Netrapalli, Sanghavi], [Hardt]

Theorem: If M is n x n and has rank r, and is C-incoherent then alternating minimization approximately recovers M from

$$Cnr^2 \frac{\|M\|_F^2}{\sigma_r^2} \text{ observations}$$

ALTERNATING MINIMIZATION

Repeat: $$U \leftarrow \underset{U}{\text{argmin}} \sum_{(i,j)\in\Omega} |(UV^T)_{i,j} - M_{i,j}|^2$$

$$V \leftarrow \underset{V}{\text{argmin}} \sum_{(i,j)\in\Omega} |(UV^T)_{i,j} - M_{i,j}|^2$$

[Keshavan, Montanari, Oh], [Jain, Netrapalli, Sanghavi], [Hardt]

Theorem: If M is n x n and has rank r, and is C-incoherent then alternating minimization approximately recovers M from

$$Cnr^2 \frac{\|M\|_F^2}{\sigma_r^2} \text{ observations}$$

Running time and space complexity are better

What if an adversary reveals more entries of M?

What if an adversary reveals more entries of M?

Convex program:

$$\min \|X\|_* \text{ s.t. } \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \qquad \textbf{(P)}$$

still works, it's just more constraints

What if an adversary reveals more entries of M?

Convex program:

$$\min \|X\|_* \text{ s.t. } \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \qquad \textbf{(P)}$$

still works, it's just more constraints

Alternating minimization:

Analysis completely breaks down

observed matrix is no longer good spectral approx. to M

What if an adversary reveals more entries of M?

Convex program:

$$\min \|X\|_* \text{ s.t. } \sum_{(i,j)\in\Omega} |X_{i,j} - M_{i,j}| \leq \eta \qquad \textbf{(P)}$$

still works, it's just more constraints

Alternating minimization:

Are there variants that work in semi-random models?

Summary:

- Nearly optimal algorithm for agnostically learning a high-dimensional Gaussian
- General recipe using restricted eigenvalue problems
- **Is practical, robust statistics within reach?**
- **Tension between nonconvex methods and being robust**

Thanks! Any Questions?