Stable Estimators for Fast Private Statistics

Gavin Brown

6/13/24

UNIVERSITY of WASHINGTON

Based (mostly) on two papers with subsets of:

Jon Hayase
Xiyang Liu
Sewoong Oh
Weihao Kong
Sam Hopkins
Juanky Perdomo
Adam Smith

Statistics with Differential Privacy

Mean Estimation

Given $x_1, \ldots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\mu} \approx \mu$

Covariance Estimation

Given $x_1, \ldots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\Sigma} \approx \Sigma$

Linear Regression

Given covariates $x_1, \ldots, x_n \sim \mathcal{N}(0, \Sigma)$
and labels $y_i = \langle x_i, \beta \rangle + \mathcal{N}(0, \sigma^2)$

produce $\hat{\beta} \approx \beta$

Constraint: algorithms satisfy **differential privacy**

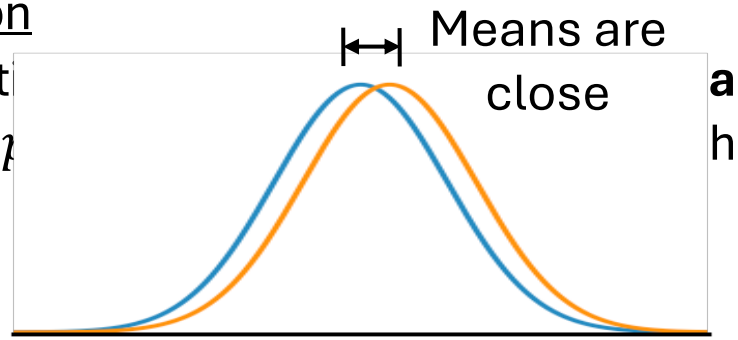
Definition

Algorithm $\mathcal{A}$ is (ε, δ)-**differentially private** if, for any data sets x and x' that differ in one person's data, we have

$$\mathcal{A}(x) \approx_{(\varepsilon,\delta)} \mathcal{A}(x')$$

Definition

Distribut **able** (written μ have



Statistics with Differential Privacy

Mean Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\mu} \approx \mu$

Covariance Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\Sigma} \approx \Sigma$

Linear Regression

Given covariates $x_1, \dots, x_n \sim \mathcal{N}(0, \Sigma)$
and labels $y_i = \langle x_i, \beta \rangle + \mathcal{N}(0, \sigma^2)$

produce $\hat{\beta} \approx \beta$

Constraint: algorithms satisfy **differential privacy**

Definition

Algorithm $\mathcal{A}$ is (ε, δ)-**differentially private** if, for any data sets x and x' that differ in one person's data, we have

$$\mathcal{A}(x) \approx_{(\varepsilon,\delta)} \mathcal{A}(x')$$

Two Analyses

Privacy guarantee: **worst-case** pair of datasets

Accuracy under distributional assumptions

Fast Algorithms for Private Statistics

Mean Estimation

Covariance Estimation

Linear Regression

Design Goals

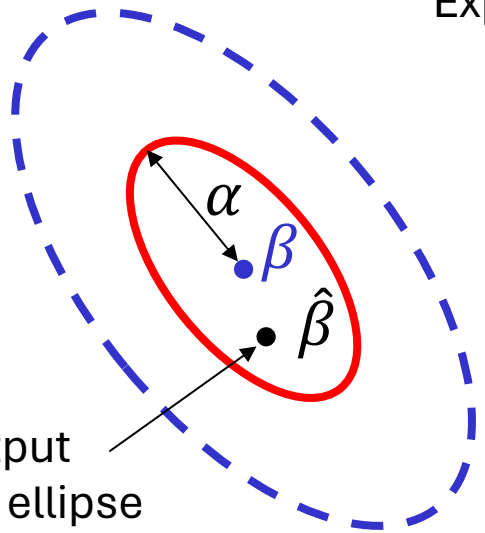
1) Low "cost of privacy"
   - Error comparable to empirical estimator
   - Computation comparable to empirical estimator
2) "Unrestricted," no assumptions on μ, Σ, β
   - E.g., of the form $\|\mu\| \leq R$
   - This goal requires approximate DP
3) Weaker distributional assumptions
   - Empirical concentration, or subgaussian

Linear Regression Under Differential Privacy

Linear Regression

Given covariates $x_1, \ldots, x_n \sim \mathcal{N}(0, \Sigma)$
and labels $y_i = \langle x_i, \beta \rangle + \mathcal{N}(0, \sigma^2)$

produce $\hat{\beta}$ such that $\left\|\Sigma^{1/2}\left(\hat{\beta} - \beta\right)\right\| \leq \alpha$



Expected error on fresh data:

$$\mathbb{E}(y - \hat{y})^2 = \mathbb{E}\left(x^T\left(\beta - \hat{\beta}\right)\right)^2 + \sigma^2$$
$$= \mathbb{E}\left(\beta - \hat{\beta}\right)^T xx^T\left(\beta - \hat{\beta}\right) + \sigma^2$$

Without privacy: ordinary least squares!

$$\beta_{\text{ols}} = (X^T X)^{-1} X^T y$$

no condition number

$$\kappa \stackrel{\text{def}}{=} \frac{\lambda_{\max}(\Sigma)}{\lambda_{\min}(\Sigma)}$$

in d dimensions:

$$n \gtrsim \frac{\sigma^2 d}{\alpha^2} \stackrel{\text{whp}}{\Longrightarrow} \| \Sigma^{1/2} (\beta_{\text{ols}} - \beta) \| \leq \alpha$$

not trivial with differential privacy

prior work needs $n \gtrsim d^{3/2}, \text{poly}(\kappa)$ or exponential time

Warm-up: Privately computing a mean

Input: samples $x_1, \ldots, x_n \in \mathbb{R}$

Attempt 1

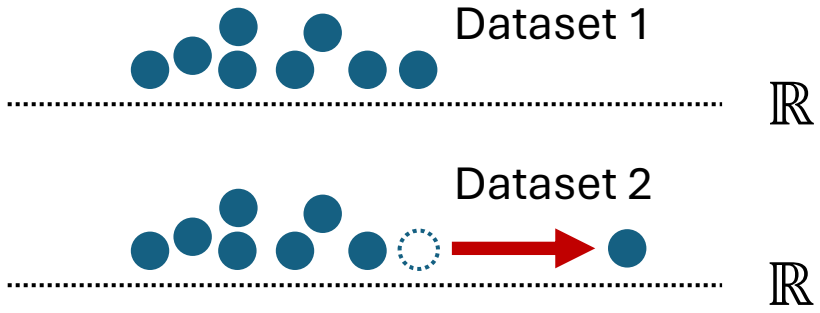
Compute $\mu_x = \frac{1}{n}\sum_i x_i$

Release $\mu_x + Z$

Random noise, independent of x

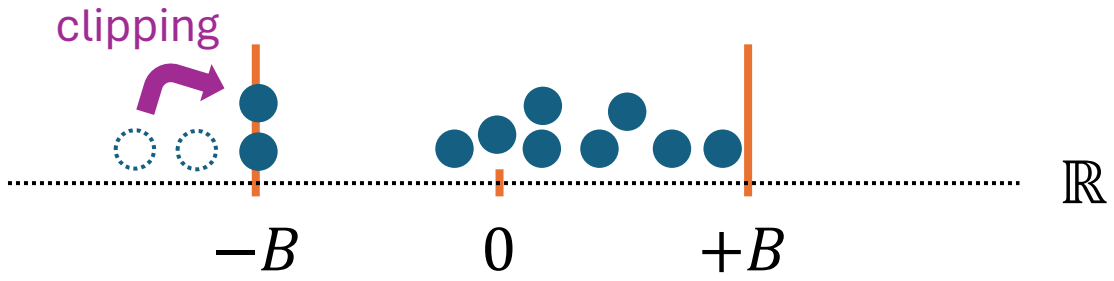
Not private!

Empirical mean highly sensitive



Attempt 2

Classic DP approach

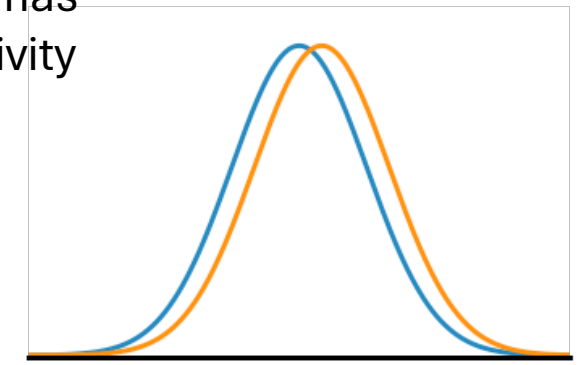


After clipping:

empirical mean has bounded sensitivity

Roughly, need

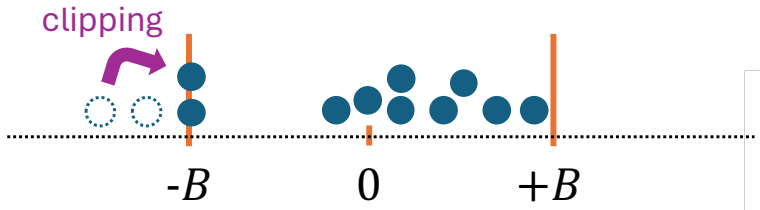
$$Z \approx_{\varepsilon,\delta} \frac{2B}{n} + Z$$

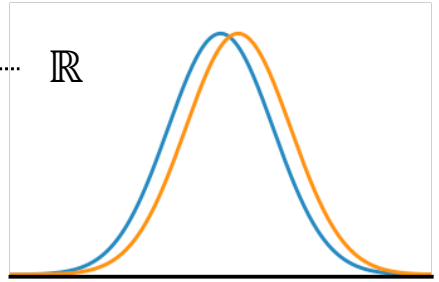


Our blueprint for private algorithms

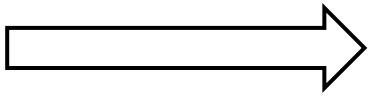
Algorithmic Template:

1. Remove outliers
2. Compute empirical estimate
3. Add noise





stronger notions of *outlier*
+
improved filtering algorithms



tight error guarantees

- Results

Outlier Filtering and How to Use It

Various Stabilizing Techniques

from other papers

Key Subroutine: Stable Leverage Filtering

from our work

Our Results (Simple Version)

Mean Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\mu}$ with

$$\left\|\Sigma^{-1/2}(\hat{\mu} - \mu)\right\| \leq \alpha$$

Mahalanobis distance

Covariance Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\Sigma}$ with

i. $(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$

ii. $\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \leq \alpha$

e.g., learn a Gaussian in TV distance

Linear Regression

Given covariates $x_1, \dots, x_n \sim \mathcal{N}(0, \Sigma)$ and labels $y_i = \langle x_i, \beta \rangle + \mathcal{N}(0, \sigma^2)$

produce $\hat{\beta}$ with

$$\left\|\Sigma^{1/2}\left(\hat{\beta} - \beta\right)\right\| \leq \alpha$$

e.g., as preconditioner

Our Results (Simple Version)

Mean Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\mu}$ with
$\left\|\Sigma^{-1/2}(\hat{\mu} - \mu)\right\| \leq \alpha$

Covariance Estimation

Given $x_1, \dots, x_n \sim \mathcal{N}(\mu, \Sigma)$

produce $\hat{\Sigma}$ with

i. $(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$

ii. $\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \leq \alpha$

Linear Regression

Given covariates $x_1, \dots, x_n \sim \mathcal{N}(0, \Sigma)$
and labels $y_i = \langle x_i, \beta \rangle + \mathcal{N}(0, \sigma^2)$

produce $\hat{\beta}$ with
$\left\|\Sigma^{1/2}\left(\hat{\beta} - \beta\right)\right\| \leq \alpha$

For these tasks, we give private algorithms with nearly optimal sample complexity,

requiring practical, lightweight computation.

Our Results (Hairy Version)

Mean Estimation

BHS23

Let $\mathcal{P}$ be subgaussian with mean $\mu \in \mathbb{R}^d$ and covariance $\Sigma \in \mathbb{R}^{d \times d}$.

Given n i.i.d. samples from $\mathcal{P}$, with high constant probability we return $\hat{\mu}$ such that

$$\left\| \Sigma^{-1/2} \left(\hat{\mu} - \mu \right) \right\| \leq \alpha$$

as long as

$$n \gtrsim \frac{d}{\alpha^2} + \frac{d\sqrt{\log 1/\delta}}{\alpha\varepsilon} + \frac{d \log 1/\delta}{\varepsilon}$$

error due to sampling

error due to privacy

cost to get started

prior work needs $n \gtrsim d^{3/2}, \mathrm{poly}(\kappa)$ or exponential time

Our Results (Really Hairy Version)

Estimation Task	Error	Nonpriv.	"Cost of Privacy" / Rate	"Cost to Start"	
Means BHS23	$\left\|\Sigma^{-1/2}(\hat{\mu}-\mu)\right\| \le \alpha$	$\frac{d}{\alpha^2}$	$\frac{d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	Concurrent with DHK23
Covariance BHS23	$(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$	$\frac{d}{\alpha^2}$	$\frac{d^{3/2}\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	First with $n = o(d^2)$ Match weaker KLSU18
	$\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \le \alpha$	$\frac{d^2}{\alpha^2}$	$\frac{d^2\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	Matches AL22, TCKMS22 nearly HKMN23
Linear Regression BHHKLOPS24	$\left\|\Sigma^{1/2}(\hat{\beta}-\beta)\right\| \le \alpha$	$\frac{\sigma^2 d}{\alpha^2}$	$\frac{\sigma d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d\,(\log 1/\delta)^2}{\varepsilon^2}$	First efficient with $n = o(d^{3/2})$ & no condition #

Our Results (Really Hairy Version)

Delete red text to match lower bounds

Estimation Task	Error	Nonpriv.	"Cost of Privacy" / Rate	"Cost to Start"
Means BHS23	$\left\|\Sigma^{-1/2}(\hat{\mu}-\mu)\right\| \le \alpha$	$\frac{d}{\alpha^2}$	$\frac{d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$
Covariance BHS23	$(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$	$\frac{d}{\alpha^2}$	$\frac{d^{3/2}\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$
	$\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \le \alpha$	$\frac{d^2}{\alpha^2}$	$\frac{d^2\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$
Linear Regression BHHKLOPS24	$\left\|\Sigma^{1/2}(\hat{\beta}-\beta)\right\| \le \alpha$	$\frac{\sigma^2 d}{\alpha^2}$	$\frac{\sigma d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d\,(\log 1/\delta)^2}{\varepsilon^2}$

Our Results (Really Hairy Version)

Omit dependence on ε, δ

Estimation Task	Error	Nonpriv.	"Cost of Privacy" / Rate	"Cost to Start"	Time Needed
Means BHS23	$\left\|\Sigma^{-1/2}(\hat{\mu}-\mu)\right\| \le \alpha$	$\frac{d}{\alpha^2}$	$\frac{d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	$nd + d^3$
Covariance BHS23	$(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$	$\frac{d}{\alpha^2}$	$\frac{d^{3/2}\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	$nd^{\omega-1}$
	$\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \le \alpha$	$\frac{d^2}{\alpha^2}$	$\frac{d^2\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d \log 1/\delta}{\varepsilon}$	
Linear Regression BHHKLOPS24	$\left\|\Sigma^{1/2}(\hat{\beta}-\beta)\right\| \le \alpha$	$\frac{\sigma^2 d}{\alpha^2}$	$\frac{\sigma d\sqrt{\log 1/\delta}}{\alpha\varepsilon}$	$\frac{d\,(\log 1/\delta)^2}{\varepsilon^2}$	$nd^{\omega-1}$

Results

- Outlier Filtering and How to Use It

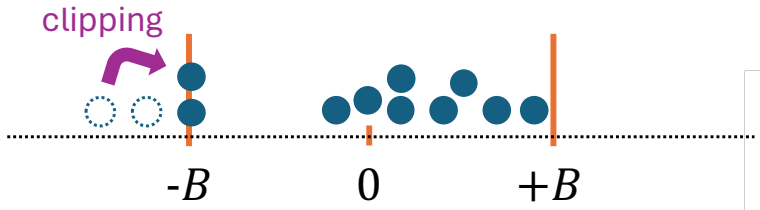
Various Stabilizing Techniques

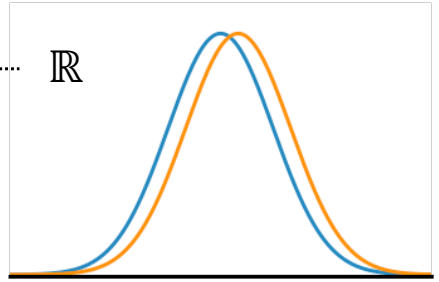
Key Subroutine: Stable Leverage Filtering

Our blueprint for private algorithms

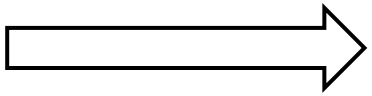
Algorithmic Template:

1. Remove outliers
2. Compute empirical estimate
3. Add noise





stronger notions of *outlier*
+
improved filtering algorithms



tight error guarantees

What is an outlier?

Task	Outlier Definition	
	"easy-to-use"	"correct"
Covariance Estimation	ℓ_2 norm $\|x_i\|_2 > B$	statistical leverage $x_i^T (X^T X)^{-1} x_i > L$
Mean Estimation	ℓ_2 norm $\|x_i\|_2 > B$	leverage & Mahalanobis distance: $\|x_i - \mu_x\|_{\hat{\Sigma}} > B$
Linear Regression	covariates $\|x_i\|_2 > B$ labels $\|y_i\| > C$	leverage & residuals: $\|y_i - x_i^T \beta_{\text{ols}}\| > R$

Three goals for outlier removal

Filtering algorithm produces weights:

$$w : \mathcal{X}^n \to [0,1]^n$$

0/1 output means hard removal

Soundness

support of $w(x)$ contains no outliers

Completeness

x has no outliers $\Rightarrow$ $w(x) = \vec{1}$

Stability

x, x' adjacent $\Rightarrow$ $\|w(x) - w(x')\|_1 = O(1)$

Our Algorithm for Private Linear Regression

Filtering algorithm produces weights:

$$w : \mathcal{X}^n \to [0,1]^n$$

Soundness | Completeness | Stability

How do we use this subroutine?

Algorithmic Template:

1. Remove outliers
2. Compute empirical estimate
3. Add noise

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$

"propose-test-release" we will ignore

4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$

weighted OLS & covariance

7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Linear Regression: Algorithm and Analysis

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$

4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$

7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Three components to analysis:

Privacy

On worst-case neighboring (X, y) and (X', y'), output distributions indistinguishable

Accuracy

On (X, y) from correct distribution, low error w.h.p.

Running Time

Linear Regression: Algorithm and Analysis

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

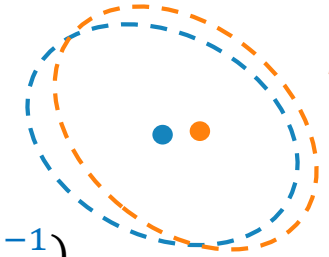
1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$
4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$
7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Privacy

(X, y) and (X', y') differ in one entry

Get weights $w = w(X, y)$ and $w' = w(X', y')$

Two Gaussians:



$\beta_{w'} + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_{w'}^{-1})$

$\beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Two Steps:

Soundness Stability

1. Identifiability: weights close in ℓ_1 and outlier-free ⇒ similar parameters
2. Indistinguishability: similar parameters ⇒ indistinguishable Gaussians

Linear Regression: Algorithm and Analysis

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$
4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$
7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Accuracy

Completeness

When (X, y) has no outliers, $w = \vec{1}$

$$S_w = X^T X \quad \text{and} \quad \beta_w = \beta_{\text{ols}}$$

Want to show $\|\Sigma^{1/2}(\beta^* - \hat{\beta})\|$ small.

Triangle inequality:

Empirical error

$$\|\Sigma^{1/2}(\hat{\beta} - \beta^*)\| \leq \|\Sigma^{1/2}(\beta^* - \beta_{\text{ols}})\| + \|\Sigma^{1/2}(\beta_{\text{ols}} - \hat{\beta})\|$$

$$\beta_{\text{ols}} - \hat{\beta} = \mathcal{N}(0, c^2 (X^T X)^{-1})$$

$$= c\,(X^T X)^{-1/2} \cdot \mathcal{N}(0, \mathbb{I})$$

Summary: Using Stable Filtering

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$
4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$
7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Privacy

Soundness

support of $w(x)$ contains no outliers

Stability

x, x' adjacent $\Rightarrow$ $\|w(x) - w(x')\|_1 = O(1)$

Accuracy

Completeness

x has no outliers $\Rightarrow$ $w(x) = \vec{1}$

Aside #1: Mean and Covariance Estimation

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$

4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$

7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

For mean estimation, filter Mahalanobis outliers

For covariance estimation, sample Wishart distribution

Aside #2: What We Knew in 2021

Algorithm: Private Linear Regression

Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$

Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$

4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$

7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Replace with exhaustive search over small subsets

Leads to exponential-time approach in **B**GSUZ21

Similar privacy and accuracy analysis

Only weak stability:

$$\|w(x) - w(x')\|_1 = O(1/\varepsilon)$$

Results

Outlier Filtering and How to Use It

Various Stabilizing Techniques

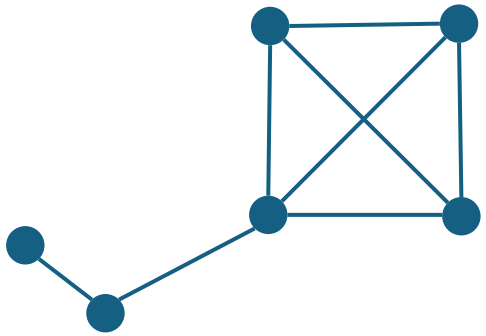
Key Subroutine: Stable Leverage Filtering

Survey

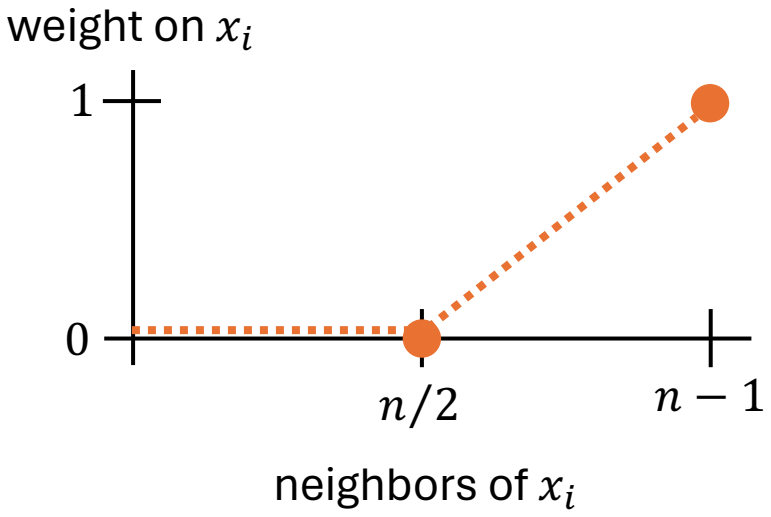
Private means in ℓ_2 with "FriendlyCore"

Tsfadia, Cohen, Kaplan, Mansour, Stemmer 22

dataset $x_1, \ldots, x_n$
$\Leftrightarrow$
graph on n vertices;
edge if $\|x_i - x_j\|_2 \leq B$



key idea: assign weight to points based on degree



Soundness

two points with weight
have common neighbor
$\therefore$ diameter $\leq 2B$

Completeness

on complete graph,
weights all 1

Stability

change point i^*
degrees can change ± 1
weights can change $\pm\frac{2}{n}$

$\|w - w'\|_1 \leq 3$

Variation 1: learning a preconditioner via cores

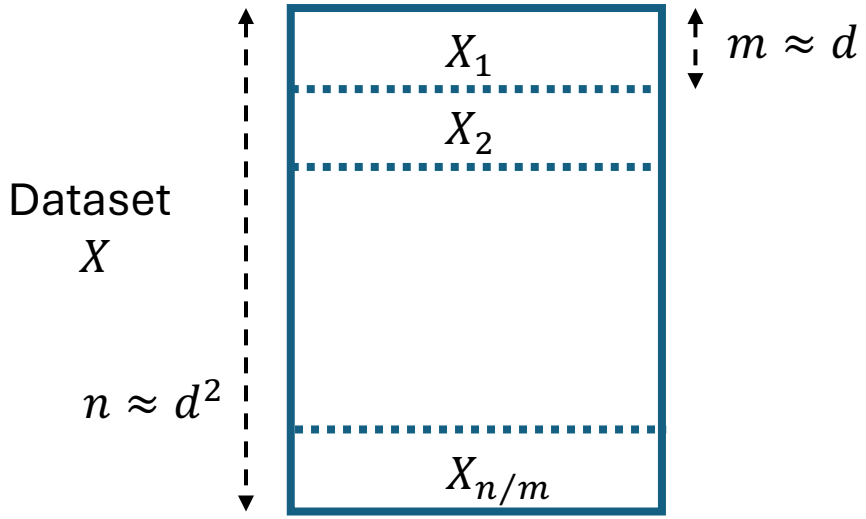
Independently, Ashtiani and Liaw 22 landed on similar idea for learning unrestricted Gaussians in TV.

Key subroutine uses $\approx d^2$ samples to learn $\tilde{\Sigma}$ with $0.9\,\Sigma \preccurlyeq \tilde{\Sigma} \preccurlyeq 1.1\,\Sigma$

with this, can use d^2 samples and KLSU18 to learn in Frobenius norm

Previously: vertex ⇔ data point

Now: vertex ⇔ covariance of a subset



Distance "metric"

$$d_{\mathrm{psd}}(\Sigma_1, \Sigma_2) \approx \left\| \Sigma_1^{-1/2} \Sigma_2 \Sigma_1^{-1/2} - \mathbb{I} \right\|_2$$

Symmetric term, pseudoinverse, zero if columnspace mismatch

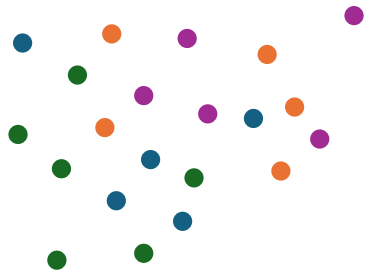
Cores: applications and variations

- TCKMS22 apply FriendlyCore to clustering
  - Also provide CDP variant
- Use it in covariance-aware mean estimation
  - With private $\tilde{\Sigma}$, use metric $\left\|\tilde{\Sigma}^{-1/2}\left(x_i - x_j\right)\right\|_2$
  - With **stable** (but not private) estimate $\hat{\Sigma}_{\mathrm{st}}$, need new ideas

- Speedups
  - All-pairs distances needs $\Omega(n^2)$
  - Select random reference set of size $O(\log n/\delta)$
  - Some variants amenable to sketching

Stabilizing the mean via random partitions

Duchi, Haque, and Kuditipudi 23
Stabilize the mean



Randomly partition/color
Each of size $b \approx \log n/\delta$

Compute diameter of each partition

Toss all points from any large partition

With fewer samples, stability
$\|w - w'\|_1 \leq \log n/\delta$

Soundness We have $\lfloor n/b \rfloor$ groups $S_1, S_2, \ldots, S_{\lfloor n/b \rfloor}$

Lemma. With probability $1 - n^2 2^{-b}$ over the partition, for all $J \subseteq \lfloor n/b \rfloor$ we have

$$\operatorname{diam}\left(\bigcup_{j \in J} S_j\right) \leq 2 \cdot \max_{j \in J} \operatorname{diam}(S_j)$$

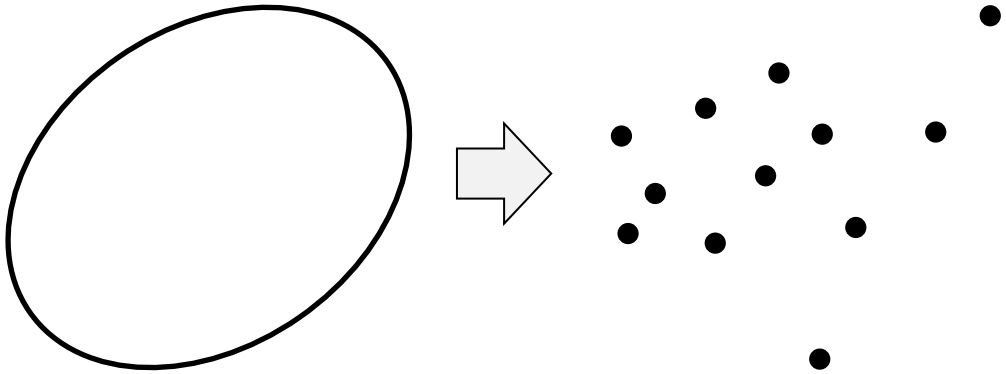
Survey

From stable covariance to private covariance

"Gaussian sampling mechanism" from AKTVZ22

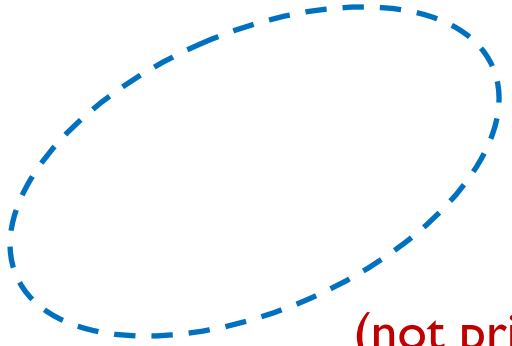
True covariance Σ

Private data $x_i \sim \mathcal{N}(0, \Sigma)$



Leverage filtering

Stable covariance $\hat{\Sigma}_{\mathrm{st}}$



(not private)

Two samples ~~One sample~~

~~x_i~~ $\sim \mathcal{N}(0, \hat{\Sigma}_{\mathrm{st}})$

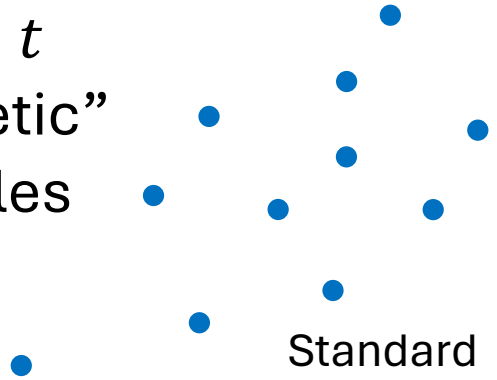
z_1, z_2

(private!)

increase n for same ε, δ

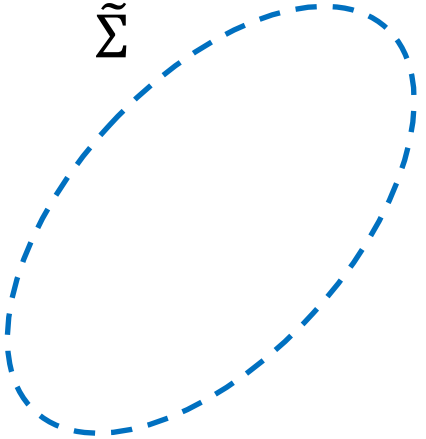
Task	t required	n required
mean	1	d
covariance (spectral)	$\Omega(d)$	$d \cdot \sqrt{t} = d^{3/2}$
covariance (Frobenius)	$\Omega(d^2)$	$d \cdot \sqrt{t} = d^2$

draw t "synthetic" samples



Standard arguments: $\Sigma \approx \hat{\Sigma}_{\mathrm{st}} \approx \tilde{\Sigma}$

Private covariance $\tilde{\Sigma}$



Results

Outlier Filtering and How to Use It

Various Stabilizing Techniques

Key Subroutine: Stable Leverage Filtering

- Overview of Algorithm
- Basics of Analysis

Remainder of talk: leverage filtering

Want to remove high-leverage points

in general: $\ell(i) = x_i^T (X^T X)^{-1} x_i$

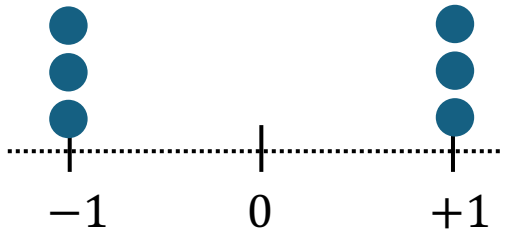
one dimension: $\ell(i) = \frac{x_i^2}{\sum_j x_j^2}$

Outliers have $\ell(i) \gg \frac{d}{n}$

Call $\sum_j x_j^2$, $X^T X$ the "(co)variance"

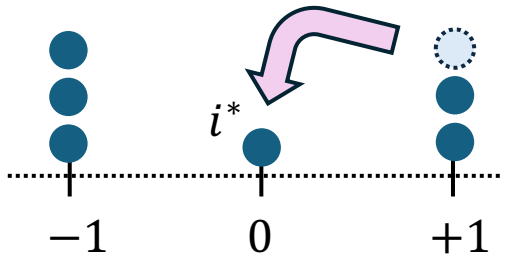
Why isn't this trivial?

Dataset 1
evenly split between ± 1



$\forall i, \ell(i) = \frac{1}{n}$

Dataset 2
set $x_{i^*} = 0$



worst case

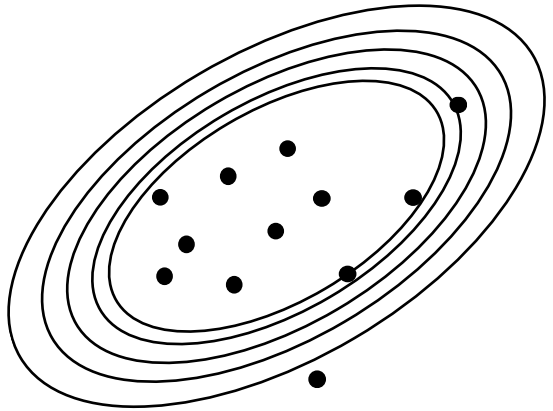
$\forall i \neq i^*, \ell(i) = \frac{1}{n-1}$

All but one point now outliers!

Introduce leverage filtering algorithm

Basics ideas of analysis

Stable Leverage Filtering



Algorithm: StableLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L_0 \in [0,1]$
level count $k \in \mathbb{N}$

Output: $\vec{w} \in [0,1]^n$

1. For $j = k, k-1, \dots, 1$:
   i. $S_j \leftarrow \text{GreedyLeverageFilter}(x, \ e^{jL_0} \cdot L_0)$
2. End
3. $\forall i, \ w_i \leftarrow \frac{1}{k} \sum_{j=1}^{k} 1\{i \in S_j\}$
4. Return $\vec{w}$

For residual filtering or Mahalanobis filtering, swap in different greedy alg.

Algorithm: GreedyLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L \in [0,1]$

Output: $S \subseteq [n]$

1. $S \leftarrow [n]$
2. Repeat
   1. $S_{\text{out}} \leftarrow \{i \in S : \ell_S(i) > L\}$
   2. $S \leftarrow S \setminus S_{\text{out}}$
3. Until $S_{\text{out}} = \emptyset$
4. Return S

$$\ell_S(i) \stackrel{\text{def}}{=} x_i^T \left(X_S^T X_S\right)^{-1} x_i$$

Check our three goals

Algorithm: StableLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L_0 \in [0,1]$
level count $k \in \mathbb{N}$

Output: $\vec{w} \in [0,1]^n$

1. For $j = k, k-1, \dots, 1$:
   i. $S_j \leftarrow \text{GreedyLeverageFilter}(x, \ e^{jL_0} \cdot L_0)$
2. End
3. $\forall i, \ w_i \leftarrow \frac{1}{k} \sum_{j=1}^{k} 1\{i \in S_j\}$
4. Return $\vec{w}$

Soundness

support of $w(x)$ contains no outliers

Completeness

x has no outliers $\Rightarrow$ $w(x) = \vec{1}$

Stability

x, x' adjacent $\Rightarrow$ $\|w(x) - w(x')\|_1 = O(1)$

Results

Outlier Filtering and How to Use It

Various Stabilizing Techniques

Key Subroutine: Stable Leverage Filtering

- Overview of Algorithm
- Basics of Analysis

Greedy finds good sets

Definition: Call $S \subseteq [n]$ an ***L*-good subset** if $\forall i \in S,\ \ell_S(i) \leq L.$

$$\ell_S(i) \stackrel{\text{def}}{=} x_i^T \left(X_S^T X_S\right)^{-1} x_i$$

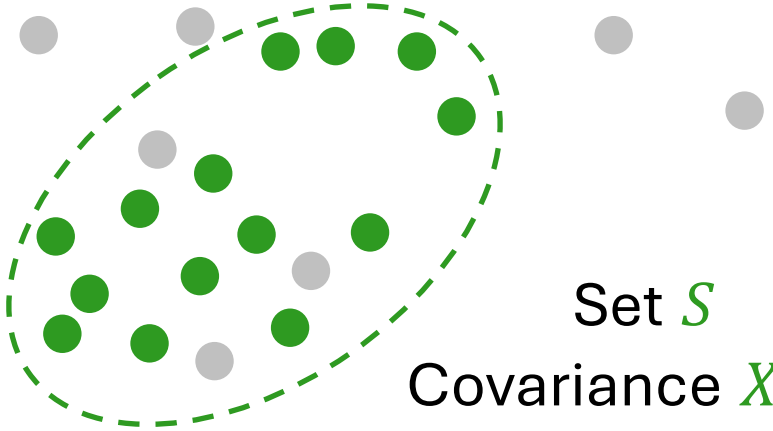
Data set x

Greedy finds good sets

Definition: Call $S \subseteq [n]$ an ***L*-good subset** if $\forall i \in S,\ \ell_S(i) \leq L$.

$$\ell_S(i) \stackrel{\text{def}}{=} x_i^T \left(X_S^T X_S\right)^{-1} x_i$$

Data set x



Set S

Covariance $X_S X_S^T$

Ellipse

$\{z : z^T \left(X_S^T X_S\right)^{-1} z = L\}$

No points beyond L threshold

$\stackrel{\text{def}}{=}$

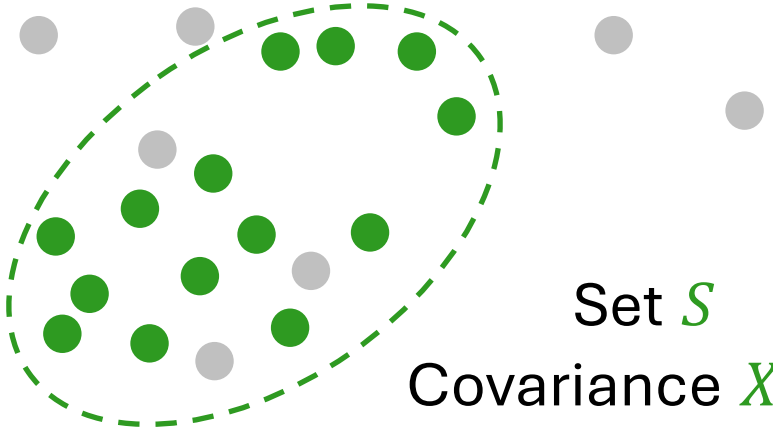
S is L-good for x

Greedy finds good sets

Definition: Call $S \subseteq [n]$ an ***L*-good subset** if $\forall i \in S,\ \ell_S(i) \leq L$.

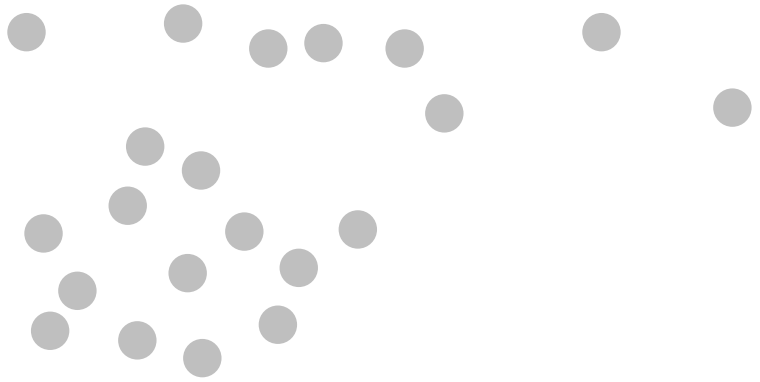
$$\ell_S(i) \stackrel{\text{def}}{=} x_i^T \left(X_S^T X_S\right)^{-1} x_i$$

Data set x



Set S

Covariance $X_S X_S^T$



Ellipse

$\{z : z^T \left(X_S^T X_S\right)^{-1} z = L\}$

No points beyond L threshold

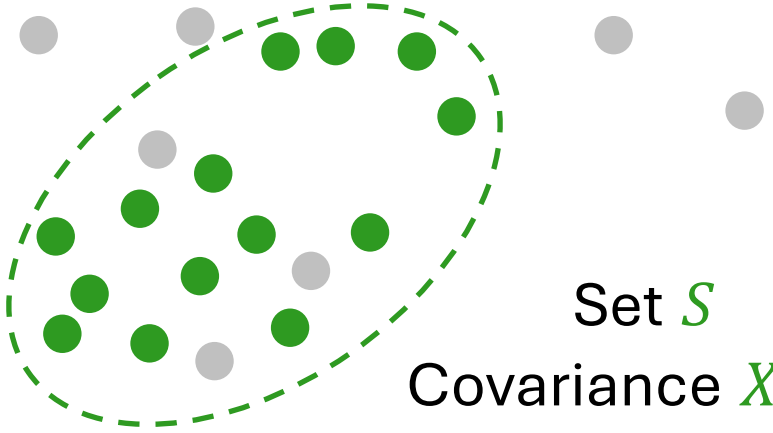
$\stackrel{\text{def}}{=}$

S is L-good for x

Greedy finds good sets

Definition: Call $S \subseteq [n]$ an ***L*-good subset** if $\forall i \in S,\ \ell_S(i) \leq L$.

Data set x



Set S

Covariance $X_S X_S^T$

Ellipse

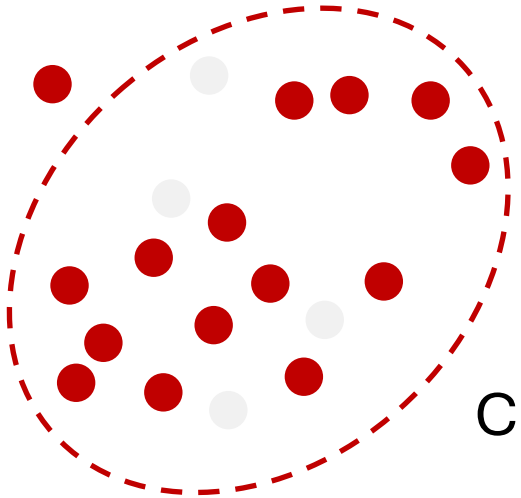
$\{z : z^T (X_S^T X_S)^{-1} z = L\}$

No points beyond L threshold

$\stackrel{\text{def}}{=}$

S is L-good for x

$\ell_S(i) \stackrel{\text{def}}{=} x_i^T (X_S^T X_S)^{-1} x_i$



Set S'

Covariance $X_{S'} X_{S'}^T$

Outliers beyond L threshold

$\stackrel{\text{def}}{=}$

S' is **not** L-good for x

Greedy finds good sets

Definition: Call $S \subseteq [n]$ an ***L*-good subset** if $\forall i \in S,\ \ell_S(i) \leq L$.

Lemma: Fix $x \in \mathbb{R}^n$, $L > 0$. Let $S \leftarrow \text{GreedyLeverageFilter}(x,\ L)$. If T is L-good, then $T \subseteq S$.

Corollary: $\text{GreedyLeverageFilter}(x,\ L)$ finds the unique largest L-good subset.

Quick Proof: Only remove PSD elements from $X_S X_S^T$. Leverage is non-decreasing under removal. ■

$$\ell_S(i) \stackrel{\text{def}}{=} x_i^T \left(X_S^T X_S\right)^{-1} x_i$$

Algorithm: GreedyLeverageFilter

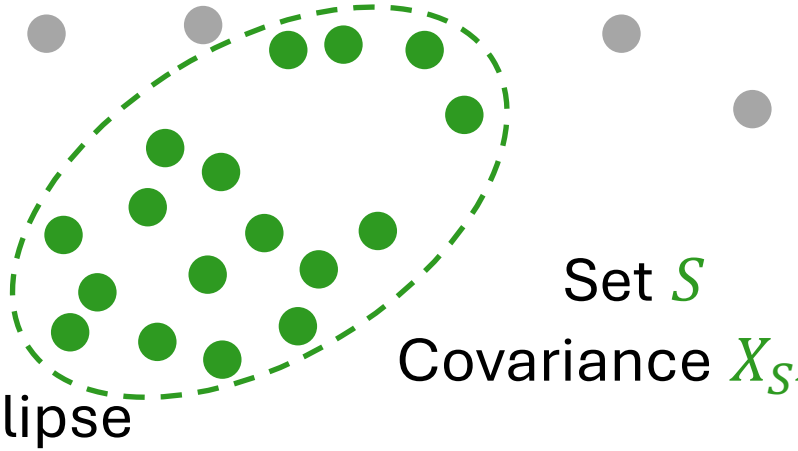
Inputs: data $x = (x_1, \ldots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L \in [0,1]$

Output: $S \subseteq [n]$

1. $S \leftarrow [n]$
2. Repeat
   1. $S_{\text{out}} \leftarrow \{i \in S : \ell_S(i) > L\}$
   2. $S \leftarrow S \setminus S_{\text{out}}$
3. Until $S_{\text{out}} = \emptyset$
4. Return S

The bad news: greedy not stable

Dataset x has a L-good subset:

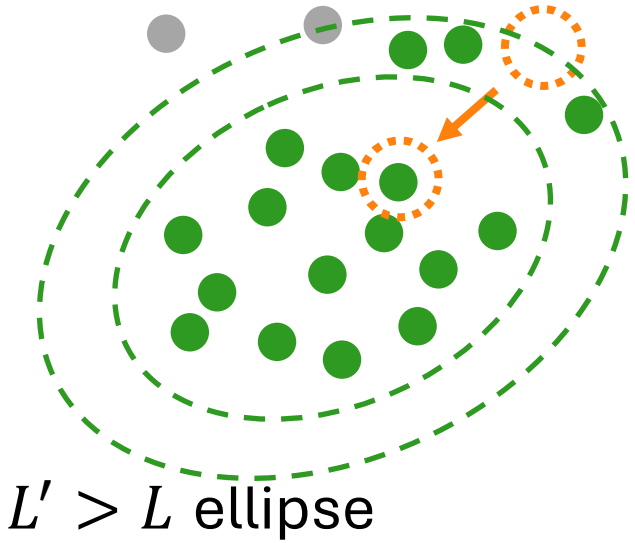


L ellipse

Set S

Covariance $X_S X_S^T$

Dataset x' differs in one point, i^*



$L' > L$ ellipse

New covariance!

S is not L-good for x'

☹️

Many outliers!

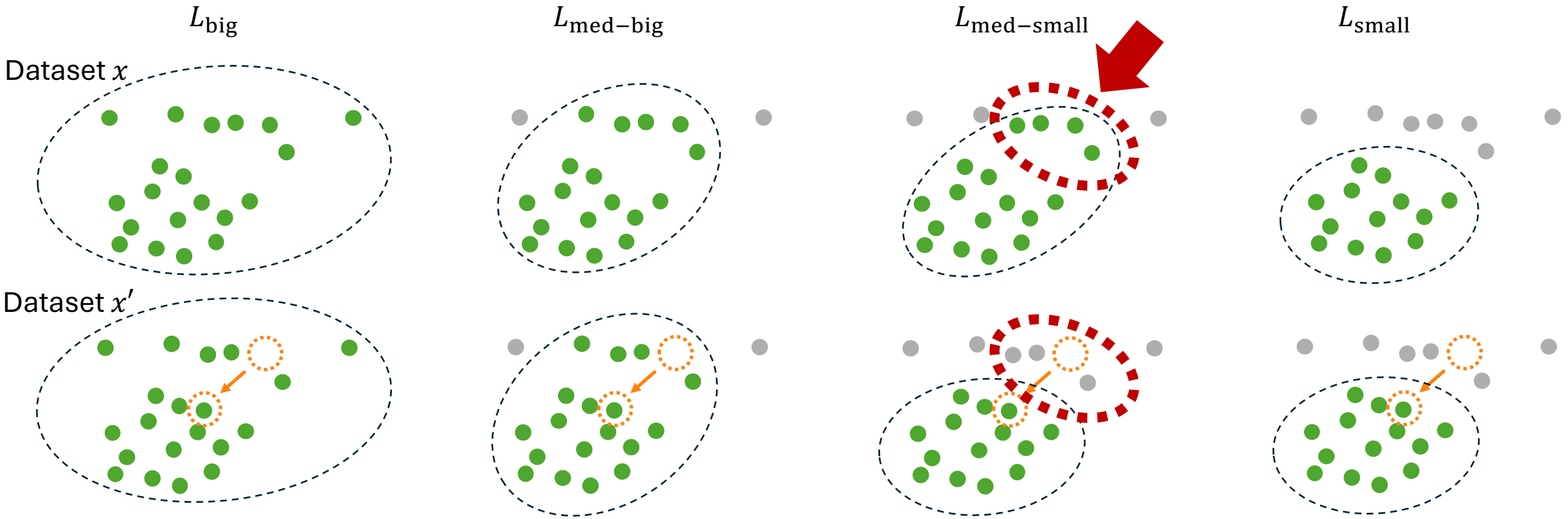
😱

Good with slightly larger outlier threshold

🥳

Stabilize over multiple thresholds

Good sets are stable “on average” across thresholds



Back to the algorithm

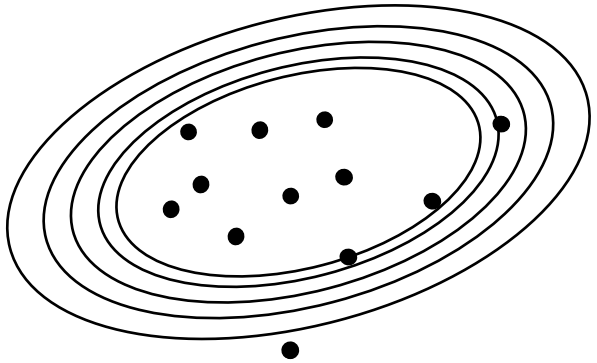
Algorithm: StableLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L_0 \in [0,1]$
level count $k \in \mathbb{N}$

Output: $\vec{w} \in [0,1]^n$

1. For $j = k, k-1, \dots, 1$:
   i. $S_j \leftarrow \text{GreedyLeverageFilter}(x, \ e^{jL_0} \cdot L_0)$
2. End
3. $\forall i, \ w_i \leftarrow \frac{1}{k} \sum_{j=1}^{k} 1\{i \in S_j\}$
4. Return $\vec{w}$

Averaging over good subsets



Why this series of thresholds?

Lemma: If $S \subseteq [n]$ is L-good, then $S \setminus \{i\}$ is $(1 + L)L$-good.

Think: $L \approx \frac{d}{n}$

Start with: L_0

$$L_1 = (1 + L_0) \cdot L_0 \approx e^{L_0} \cdot L_0$$
$$\vdots$$
$$L_j = (1 + L_0) \cdot L_{j-1} \approx e^{jL_0} \cdot L_0$$

Summary: analysis of leverage filtering

Algorithm: StableLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L_0 \in [0,1]$
level count $k \in \mathbb{N}$

Output: $\vec{w} \in [0,1]^n$

1. For $j = k, k-1, \dots, 1$:
   i. $S_j \leftarrow \text{GreedyLeverageFilter}(x, \quad e^{jL_0} \cdot L_0)$
2. End
3. $\forall i, \quad w_i \leftarrow \frac{1}{k} \sum_{j=1}^{k} 1\{i \in S_j\}$
4. Return $\vec{w}$

Soundness

support of $w(x)$ contains no outliers

Completeness

x has no outliers ⇒ $w(x) = \vec{1}$

Stability

x, x' adjacent ⇒ $\|w(x) - w(x')\|_1 = O(1)$

To extend the approach, swap in different greedy algorithm.

Algorithm: GreedyLeverageFilter

Inputs: data $x = (x_1, \dots, x_n) \in \mathbb{R}^{n \times d}$
leverage threshold $L \in [0,1]$

Output: $S \subseteq [n]$

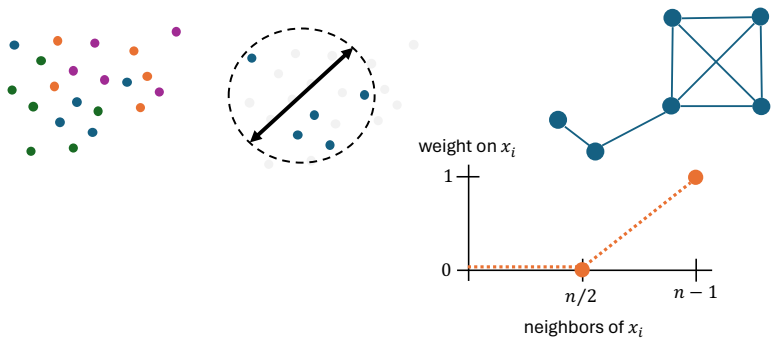
1. $S \leftarrow [n]$
2. Repeat
   1. $S_{\text{out}} \leftarrow \{i \in S : \ell_S(i) > L\}$
   2. $S \leftarrow S \setminus S_{\text{out}}$
3. Until $S_{\text{out}} = \emptyset$
4. Return S

Summary

stronger notions of outlier
+
improved filtering algorithms

⇒

tight error guarantees



Algorithm: Private Linear Regression
Inputs: data (X, y)
privacy parameters $\varepsilon, \delta > 0$
Output: $\hat{\beta} \in \mathbb{R}^d$

1. $v \leftarrow \text{StableLeverageFilter}(X)$
2. $w \leftarrow \text{StableResidualFilter}(X, y, v)$
3. $\text{CheckOutlierCount}(\cdot)$

4. $W \leftarrow \text{diag}(w)$
5. $S_w \leftarrow X^T W X$
6. $\beta_w \leftarrow (X^T W X)^{-1} X^T W y$

7. Return $\hat{\beta} = \beta_w + \mathcal{N}(0, c_{\varepsilon,\delta}^2 \cdot S_w^{-1})$

Privacy

Soundness

support of $w(x)$ contains no outliers

Stability

x, x' adjacent ⇒ $\|w(x) - w(x')\|_1 = O(1)$

Accuracy

Utility

x has no outliers ⇒ $w(x) = \vec{1}$

Task	Error	Samples Needed	Time Needed
Mean Estimation	$\left\|\Sigma^{-1/2}(\hat{\mu} - \mu)\right\| \le \alpha$	$\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon}$	$nd + d^3$
Covariance Estimation	$(1-\alpha)\Sigma \preccurlyeq \hat{\Sigma} \preccurlyeq (1+\alpha)\Sigma$	$\frac{d}{\alpha^2} + \frac{d^{3/2}}{\alpha\varepsilon}$	$nd^{\omega-1}$
	$\left\|\Sigma^{-1/2}\,\hat{\Sigma}\,\Sigma^{-1/2} - \mathbb{I}\right\|_F \le \alpha$	$\frac{d^2}{\alpha^2} + \frac{d^2}{\alpha\varepsilon}$	
Linear Regression	$\left\|\Sigma^{1/2}(\hat{\beta} - \beta)\right\| \le \alpha$	$\frac{d}{\alpha^2} + \frac{d}{\alpha\varepsilon}$	$nd^{\omega-1}$

Thank you!

- Alabi, Daniel, Pravesh K. Kothari, Pranay Tankala, Prayaag Venkat, and Fred Zhang. "Privately estimating a Gaussian: Efficient, robust, and optimal." STOC 2023.
- Ashtiani, Hassan, and Christopher Liaw. "Private and polynomial time algorithms for learning Gaussians and beyond." COLT, 2022.
- B., Marco Gaboardi, Adam Smith, Jonathan Ullman, and Lydia Zakynthinou. "Covariance-aware private mean estimation without private covariance estimation." NeurIPS 2021.
- B., Jonathan Hayase, Samuel Hopkins, Weihao Kong, Xiyang Liu, Sewoong Oh, Juan C. Perdomo, and Adam Smith. "Insufficient Statistics Perturbation: Stable Estimators for Private Least Squares." COLT 2024.
- B., Samuel Hopkins, and Adam Smith. "Fast, sample-efficient, affine-invariant private mean and covariance estimation for subgaussian distributions." COLT 2023.
- Duchi, John, Saminul Haque, and Rohith Kuditipudi. "A pretty fast algorithm for adaptive private mean estimation." COLT 2023.
- Hopkins, Samuel, Gautam Kamath, Mahbod Majid, and Shyam Narayanan. "Robustness implies privacy in statistical estimation." STOC 2023.
- Kamath, Gautam, Jerry Li, Vikrant Singhal, and Jonathan Ullman. "Privately learning high-dimensional distributions." COLT 2019.
- Tsfadia, Eliad, Edith Cohen, Haim Kaplan, Yishay Mansour, and Uri Stemmer. "Friendlycore: Practical differentially private aggregation." ICML, 2022