Robust estimation of a mean in a multivariate Gaussian model: Part 2

Frejus, December 18, 2018

Arnak S. Dalalyan

ENSAE ParisTech / CREST

Quick Recap

General notation

We first introduce the notation that are common to all the models of contamination considered in this talk.

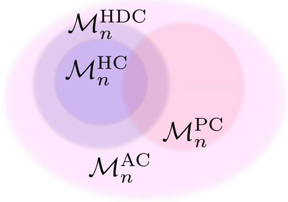
- Number of observations : n.
- Dimension of the unknown parameter $\boldsymbol{\mu}^*$: p.
- Observations $(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) \sim \boldsymbol{P}_n$.
- Number of outliers (possibly random): $s \in \{1, \ldots, n\}$.
- Set of outliers: $S \subset \{1, \ldots, n\}$.
- Proportion of outliers: $\varepsilon = \mathbf{E}[s/n] = \mathbf{E}[|S|/n]$.

Setting (informal)

Among the n observations $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, there is a small number s of outliers. If we remove the outliers, all the other $\boldsymbol{X}_i$'s are iid drawn from a reference distribution $\mathcal{N}_p(\boldsymbol{\mu}^*, \mathbf{I}_p)$.

Summary of the first lecture

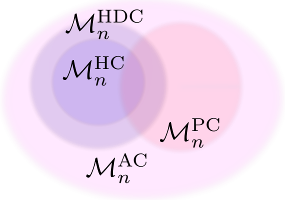
- We have introduced four models of contamination: $\mathcal{M}_n^{\square}(p, \varepsilon, \boldsymbol{\mu}^*)$.
  - Huber's Cont.: $\square = \mathrm{HC}$.
  - Huber's deterministic: $\square = \mathrm{HDC}$.
  - Parameter Cont.: $\square = \mathrm{PC}$.
  - Adversarial Cont.: $\square = \mathrm{AC}$.



Summary of the first lecture

- We have introduced four models of contamination: $\mathcal{M}_n^{\square}(p, \varepsilon, \boldsymbol{\mu}^*)$.

  - Huber's Cont.: $\square = \mathrm{HC}$.
  - Huber's deterministic: $\square = \mathrm{HDC}$.
  - Parameter Cont.: $\square = \mathrm{PC}$.
  - Adversarial Cont.: $\square = \mathrm{AC}$.

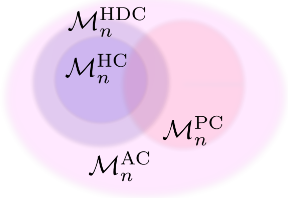


- We have defined the worst case risk $r^\star_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$ and the minimax risk $r^\star_{n,p,\varepsilon} = \inf_{\widehat{\boldsymbol{\mu}}} r^\star_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$.

Summary of the first lecture

- We have introduced four models of contamination: $\mathcal{M}_n^{\square}(p, \varepsilon, \boldsymbol{\mu}^*)$.

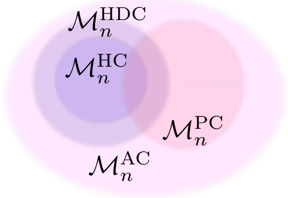
  - Huber's Cont.: $\square = \mathrm{HC}$.
  - Huber's deterministic: $\square = \mathrm{HDC}$.
  - Parameter Cont.: $\square = \mathrm{PC}$.
  - Adversarial Cont.: $\square = \mathrm{AC}$.



- We have defined the worst case risk $r^{\star}_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$ and the minimax risk $r^{\star}_{n,p,\varepsilon} = \inf_{\widehat{\boldsymbol{\mu}}} r^{\star}_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$.
- We have seen that $\forall\, \varepsilon < 1/3 - \square$, we have $r^{\star}_{n,p,\varepsilon} \asymp \boxed{\frac{p}{n} + \varepsilon^2}$.
- This minimax rate is obtained by Tukey's median.

Summary of the first lecture

- We have introduced four models of contamination: $\mathcal{M}_n^{\square}(p, \varepsilon, \boldsymbol{\mu}^*)$.
  - Huber's Cont.: $\square = \mathrm{HC}$.
  - Huber's deterministic: $\square = \mathrm{HDC}$.
  - Parameter Cont.: $\square = \mathrm{PC}$.
  - Adversarial Cont.: $\square = \mathrm{AC}$.



- We have defined the worst case risk $r^\star_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$ and the minimax risk $r^\star_{n,p,\varepsilon} = \inf_{\widehat{\boldsymbol{\mu}}} r^\star_{n,p,\varepsilon}(\widehat{\boldsymbol{\mu}})$.
- We have seen that $\forall\, \varepsilon < 1/3 - \square$, we have $r^\star_{n,p,\varepsilon} \asymp \boxed{\frac{p}{n} + \varepsilon^2}$.
- This minimax rate is obtained by Tukey's median.

Question

What is the smallest rate of the worst-case risk that can be obtained by an estimator computable in poly$(n, p, 1/\varepsilon)$ time?

4. Robust estimation by the ellipsoid method

Worst-case risk bound

Ellipsoid method for robust estimation

Theorem 3 (Diakonikolas et al., 2016)

Let $\delta \in (0, 1/2)$. There are constants $c, C > 0$ such that, for every $\varepsilon \le c$, on a set of probability $\ge 1 - \delta$, the *ellipsoid method for robust estimation* terminates in poly$(n, p, 1/\varepsilon)$ steps and outputs a weight vector $\widehat{\boldsymbol{w}} \in [0, 1]^n$ such that the mean

$$\widehat{\boldsymbol{\mu}}_n^{Ell} = \sum_{i=1}^{n} \widehat{w}_i \boldsymbol{X}_i$$

satisfies $$\|\widehat{\boldsymbol{\mu}}_n^{Ell} - \boldsymbol{\mu}^*\|_2^2 \le C\left(\frac{p}{n} + \varepsilon^2 \log(1/\varepsilon) + \frac{\log(1/\delta)}{n}\right).$$

- Valid for $\mathcal{M}_n^{\mathrm{AC}}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Complexity of 1 step: $O(np^2)$.

Worst-case risk bound

Ellipsoid method for robust estimation

Theorem 3 (Diakonikolas et al., 2016)

Let $\delta \in (0, 1/2)$. There are constants $c, C > 0$ such that, for every $\varepsilon \le c$, on a set of probability $\ge 1 - \delta$, the *ellipsoid method for robust estimation* terminates in poly$(n, p, 1/\varepsilon)$ steps and outputs a weight vector $\widehat{\boldsymbol{w}} \in [0, 1]^n$ such that the mean

$$\widehat{\boldsymbol{\mu}}_n^{Ell} = \sum_{i=1}^{n} \widehat{w}_i \boldsymbol{X}_i$$

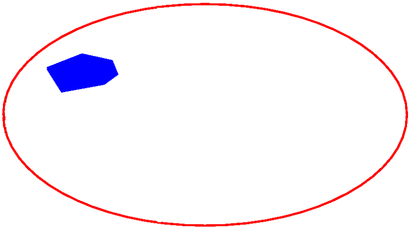
satisfies $\|\widehat{\boldsymbol{\mu}}_n^{Ell} - \boldsymbol{\mu}^*\|_2^2 \le C\bigg(\frac{p}{n} + \varepsilon^2 \underline{\log(1/\varepsilon)} + \frac{\log(1/\delta)}{n}\bigg).$

- Valid for $\mathcal{M}_n^{\mathrm{AC}}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Complexity of 1 step: $O(np^2)$.
- Extra factor $\log(1/\varepsilon)$.

- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.

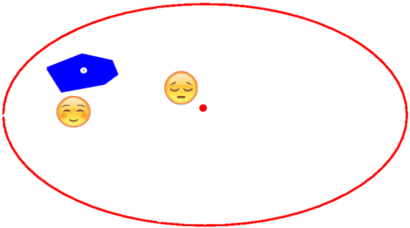
Ellipsoid Algorithm

The classic one



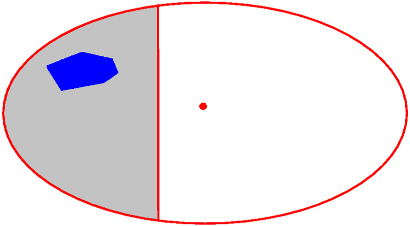
- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- One needs
  - an ellipsoid containing $\mathcal{P}$,

Ellipsoid Algorithm
The classic one



- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- One needs
  - an ellipsoid containing $\mathcal{P}$,
  - a membership oracle,

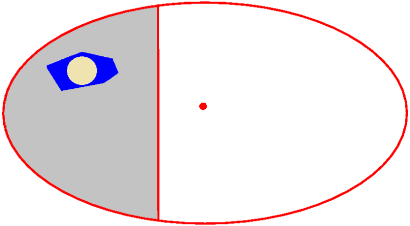
Ellipsoid Algorithm
The classic one



- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- One needs
  - an ellipsoid containing $\mathcal{P}$,
  - a membership oracle,
  - a separation oracle,

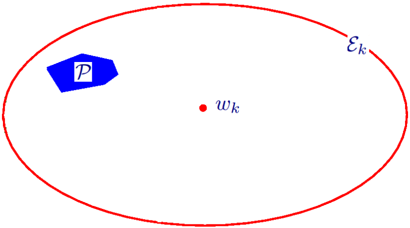
Ellipsoid Algorithm

The classic one



- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- One needs
  - an ellipsoid containing $\mathcal{P}$,
  - a membership oracle,
  - a separation oracle,
  - a lower bound on the volume of $\mathcal{P}$,

Ellipsoid Algorithm
The classic one

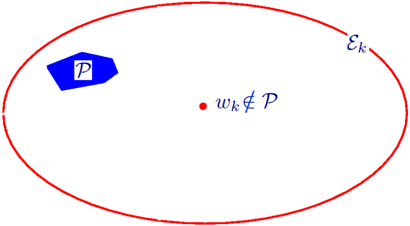


- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,

Ellipsoid Algorithm

The classic one

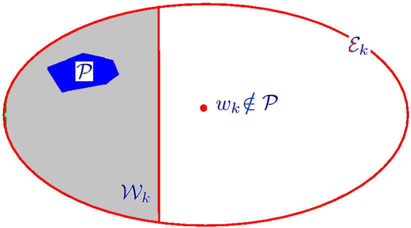


- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,

Ellipsoid Algorithm

The classic one

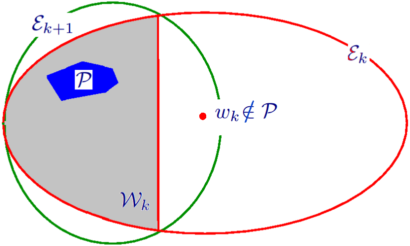


- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

Ellipsoid Algorithm

The classic one

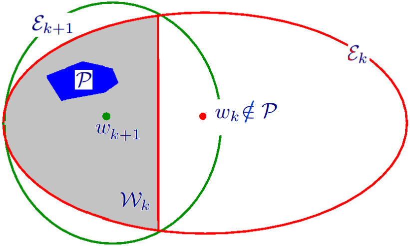


- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

Ellipsoid Algorithm

The classic one

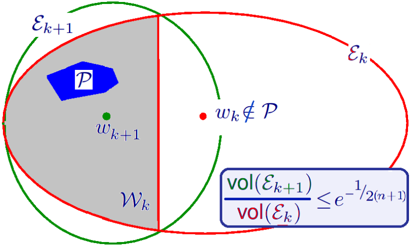


- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

- Since $\text{vol}(\mathcal{E}_{k+1}) \leq e^{-1/2(n+1)}\text{vol}(\mathcal{E}_k)$, the algo will stop after $O(n^2)$ steps,

Ellipsoid Algorithm
The classic one



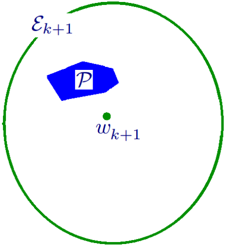
- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

- Since $\text{vol}(\mathcal{E}_{k+1}) \leq e^{-1/2(n+1)}\text{vol}(\mathcal{E}_k)$, the algo will stop after $O(n^2)$ steps, **if** $\log \textbf{vol}(\mathcal{P}) \geq -cn$.

Ellipsoid Algorithm

The classic one



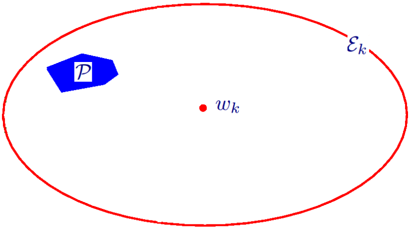
- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

- Since $\text{vol}(\mathcal{E}_{k+1}) \le e^{-1/2(n+1)}\text{vol}(\mathcal{E}_k)$, the algo will stop after $O(n^2)$ steps, **if** $\log \textbf{vol}(\mathcal{P}) \ge -cn$.

Ellipsoid Algorithm

The classic one



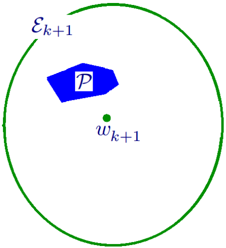
- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

- Since $\text{vol}(\mathcal{E}_{k+1}) \le e^{-1/2(n+1)}\text{vol}(\mathcal{E}_k)$, the algo will stop after $O(n^2)$ steps, **if** $\log \textbf{vol}(\mathcal{P}) \ge -cn$.

Ellipsoid Algorithm

The classic one



- Nemirovski-Yudin (1976), Chor (1976-77), Khachiyan (1979).
- The goal is to find a point w belonging to the polytope $\mathcal{P}$.
- Repeat K times:

  if the center $w_k \in \mathcal{P}$, stop and output w_k,
  else compute the minimal volume ellipsoid $\mathcal{E}_{k+1}$ containing $\mathcal{W}_k$.

- Since $\text{vol}(\mathcal{E}_{k+1}) \le e^{-1/2(n+1)}\text{vol}(\mathcal{E}_k)$, the algo will stop after $O(n^2)$ steps, **if** $\log \textbf{vol}(\mathcal{P}) \ge -cn$.

Ellipsoid method for robust estimation

Goal: use the ellipsoid algorithm for approximating the ideal weight vector $\boldsymbol{w}^*$ defined by $w_i^* = \mathbb{1}(i \in S^c)/(n-s)$, $i = 1, \ldots, n$.

- Candidate weights $\Omega = \big\{\boldsymbol{w} \in [0, \frac{1}{n-2s}]^n : \boldsymbol{w}^\top \mathbf{1}_n = 1\big\}$.
- Good weights [note that $\boldsymbol{w}^* \in \Omega^* \subset \Omega$]

$$\Omega^* = \bigg\{\boldsymbol{w} \in \Omega : \Big\| \sum_{i=1}^n w_i(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})^\top - \mathbf{I}_p \Big\|_{sp}^2 \leq \tau \bigg\}.$$

with $\tau = C\big(\frac{p}{n} + \varepsilon^2 \log(1/\varepsilon) + \frac{\log(1/\delta)}{n}\big)$.

Ellipsoid method for robust estimation

Goal: use the ellipsoid algorithm for approximating the ideal weight vector $\boldsymbol{w}^*$ defined by $w_i^* = \mathbb{1}(i \in S^c)/(n-s)$, $i = 1, \dots, n$.

- Candidate weights $\Omega = \big\{\boldsymbol{w} \in [0, \frac{1}{n-2s}]^n : \boldsymbol{w}^\top \mathbf{1}_n = 1\big\}$.
- Good weights [note that $\boldsymbol{w}^* \in \Omega^* \subset \Omega$]

$$\Omega^* = \bigg\{\boldsymbol{w} \in \Omega : \Big\| \sum_{i=1}^n w_i (\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})^\top - \mathbf{I}_p \Big\|_{sp}^2 \leq \tau \bigg\}.$$

  with $\tau = C\big(\frac{p}{n} + \varepsilon^2 \log(1/\varepsilon) + \frac{\log(1/\delta)}{n}\big)$.

- With probability $\geq 1 - \delta$,
  - $\boldsymbol{w}^* \in \Omega^*$,
  - $\forall \boldsymbol{w} \notin \Omega^*$, one can linearly separate $\boldsymbol{w}$ from $\boldsymbol{w}^*$.

Ellipsoid method for robust estimation

Goal: use the ellipsoid algorithm for approximating the ideal weight vector $\boldsymbol{w}^*$ defined by $w_i^* = \mathbb{1}(i \in S^c)/(n-s)$, $i = 1, \ldots, n$.

- Candidate weights $\Omega = \big\{\boldsymbol{w} \in [0, \frac{1}{n-2s}]^n : \boldsymbol{w}^\top \mathbf{1}_n = 1\big\}$.
- Good weights [note that $\boldsymbol{w}^* \in \Omega^* \subset \Omega$]

$$\Omega^* = \bigg\{\boldsymbol{w} \in \Omega : \Big\| \sum_{i=1}^n w_i(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})^\top - \mathbf{I}_p \Big\|_{sp}^2 \leq \tau \bigg\}.$$

  with $\tau = C\big(\frac{p}{n} + \varepsilon^2 \log(1/\varepsilon) + \frac{\log(1/\delta)}{n}\big)$.

- With probability $\geq 1-\delta$,
  - $\boldsymbol{w}^* \in \Omega^*$,
  - $\forall \boldsymbol{w} \notin \Omega^*$, one can linearly separate $\boldsymbol{w}$ from $\boldsymbol{w}^*$.
  - no lower bound on vol(Ω^*),
  - no separation between $\boldsymbol{w}$ and Ω^*.

Ellipsoid method for robust estimation

Goal: use the ellipsoid algorithm for approximating the ideal weight vector $\boldsymbol{w}^*$ defined by $w_i^* = \mathbb{1}(i \in S^c)/(n-s)$, $i = 1, \ldots, n$.

- Candidate weights $\Omega = \big\{\boldsymbol{w} \in [0, \frac{1}{n-2s}]^n : \boldsymbol{w}^\top \mathbf{1}_n = 1\big\}$.
- Good weights [note that $\boldsymbol{w}^* \in \Omega^* \subset \Omega$]

$$\Omega^* = \Big\{\boldsymbol{w} \in \Omega : \Big\| \sum_{i=1}^{n} w_i(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})(\boldsymbol{X}_i - \bar{\boldsymbol{X}}_{\boldsymbol{w}})^\top - \mathbf{I}_p \Big\|_{sp}^2 \leq \tau \Big\}.$$

with $\tau = C\big(\frac{p}{n} + \varepsilon^2 \log(1/\varepsilon) + \frac{\log(1/\delta)}{n}\big)$.

- With probability $\geq 1 - \delta$,

Does it really terminate in polynomial time ?

separate $\boldsymbol{w}$ from $\boldsymbol{w}^*$.

- no lower bound on vol(Ω^*),
- no separation between $\boldsymbol{w}$ and Ω^*.

5. Robust estimation by the spectral method

Spectral method for robust estimation

Finite sample guarantees

Theorem 4 (Lai et al., 2016)

There are constants $\alpha, c, C > 0$ such that, for every $\varepsilon \le c$, in an event of probability $\ge 1 - 1/p^\alpha$, the *spectral method for robust estimation* runs in poly$(n, p, 1/\varepsilon)$-time and outputs a vector $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ such that

$$\|\widehat{\boldsymbol{\mu}}_n^{\rm Sp} - \boldsymbol{\mu}^*\|_2^2 \le C\bigg(\frac{p(\log p)^2 \log n}{n} + \varepsilon^2 \log p\bigg).$$

- Valid for $\mathcal{M}_n^{\rm AC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity: $O(np^2)$.

Spectral method for robust estimation

Finite sample guarantees

Theorem 4 (Lai et al., 2016)

There are constants $\alpha, c, C > 0$ such that, for every $\varepsilon \le c$, in an event of probability $\ge 1 - 1/p^\alpha$, the *spectral method for robust estimation* runs in poly$(n, p, 1/\varepsilon)$-time and outputs a vector $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ such that

$$\|\widehat{\boldsymbol{\mu}}_n^{\rm Sp} - \boldsymbol{\mu}^*\|_2^2 \le C\bigg(\frac{p(\log p)^2 \log n}{n} + \varepsilon^2 \log p\bigg).$$

- Valid for $\mathcal{M}_n^{\rm AC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity: $O(np^2)$.
- Extra factors $(\log p)^2 \log n$ and $\log p$.

Spectral method for robust estimation

Finite sample guarantees

Theorem 4 (Lai et al., 2016)

There are constants $\alpha, c, C > 0$ such that, for every $\varepsilon \le c$, in an event of probability $\ge 1 - 1/p^\alpha$, the *spectral method for robust estimation* runs in poly$(n, p, 1/\varepsilon)$-time and outputs a vector $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ such that

$$\|\widehat{\boldsymbol{\mu}}_n^{\rm Sp} - \boldsymbol{\mu}^*\|_2^2 \le C\bigg(\frac{p(\log p)^2 \log n}{n} + \varepsilon^2 \log p\bigg).$$

- Valid for $\mathcal{M}_n^{\rm AC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity: $O(np^2)$.
- Extra factors $(\log p)^2 \log n$ and $\log p$.
- The decay of the probability is polynomial in p (versus exponential for other methods).

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.

2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^{\perp}$ of dimension $p/4$ using $\mathcal{D}_2$.

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ on a subspace $V_2 \subset V_1^\perp$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^{\perp}$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^{\perp}$ using $\mathcal{D}_k$.

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^{\perp}$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^{\perp}$ using $\mathcal{D}_k$.
   - Compute the median of $\{\Pi_{V_k} \boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
   - $\mathcal{D}_1' := \mathcal{D}_1 \setminus \{\boldsymbol{X} : \|\boldsymbol{X} - \text{Med}(\mathcal{D}_1)\|_2 > C\sqrt{p\log n}\}$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ on a subspace $V_2 \subset V_1^\perp$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\rm Sp}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^\perp$ using $\mathcal{D}_k$.
   - Compute the median of $\{\Pi_{V_k}\boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
   - $\mathcal{D}_1' := \mathcal{D}_1 \setminus \{\boldsymbol{X} : \|\boldsymbol{X} - \mathsf{Med}(\mathcal{D}_1)\|_2 > C\sqrt{p \log n}\}$.
   - Compute the SVD of the $\mathsf{Cov}(\mathcal{D}_1')$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^{\perp}$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^{\perp}$ using $\mathcal{D}_k$.
   - Compute the median of $\{\Pi_{V_k}\boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\text{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
   - $\mathcal{D}_1' := \mathcal{D}_1 \setminus \{\boldsymbol{X} : \|\boldsymbol{X} - \text{Med}(\mathcal{D}_1)\|_2 > C\sqrt{p \log n}\}$.
   - Compute the SVD of the $\text{Cov}(\mathcal{D}_1')$.
   - Let V_1 be the subspace of $p/2$ smallest eigenvectors.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\text{Sp}}$ on a subspace $V_2 \subset V_1^{\perp}$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\text{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^{\perp}$ using $\mathcal{D}_k$.
   - Compute the median of $\{\Pi_{V_k}\boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
    - $\mathcal{D}_1' := \mathcal{D}_1 \setminus \{\boldsymbol{X} : \|\boldsymbol{X} - \text{Med}(\mathcal{D}_1)\|_2 > C\sqrt{p \log n}\}$.
    - Compute the SVD of the $\text{Cov}(\mathcal{D}_1')$.
    - Let V_1 be the subspace of $p/2$ smallest eigenvectors.
    - Set $\Pi_{V_1}(\widehat{\boldsymbol{\mu}}^{\mathrm{Sp}}) := \text{Mean}(\Pi_{V_1}\mathcal{D}_1')$ and $\mathcal{D}_2' := \Pi_{V_1^\perp}\mathcal{D}_2$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^\perp$ of dimension $p/4$ using $\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^\perp$ using $\mathcal{D}_k$.
    - Compute the median of $\{\Pi_{V_k}\boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

Spectral method for robust estimation

The algorithm

Start by data-splitting $\boldsymbol{X}_{1:n} = \mathcal{D}_1, \ldots, \mathcal{D}_k$ with $k = [\log_2 p]$.

1. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace V_1 of dimension $p/2$ using $\mathcal{D}_1$.
   - $\mathcal{D}_1' := \mathcal{D}_1 \setminus \{\boldsymbol{X} : \|\boldsymbol{X} - \mathsf{Med}(\mathcal{D}_1)\|_2 > C\sqrt{p \log n}\}$.
   - Compute the SVD of the $\mathsf{Cov}(\mathcal{D}_1')$.
   - Let V_1 be the subspace of $p/2$ smallest eigenvectors.
   - Set $\Pi_{V_1}(\widehat{\boldsymbol{\mu}}^{\mathrm{Sp}}) := \mathsf{Mean}(\Pi_{V_1}\mathcal{D}_1')$ and $\mathcal{D}_2' := \Pi_{V_1^\perp}\mathcal{D}_2$.
2. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on a subspace $V_2 \subset V_1^\perp$ of dimension $p/4$ using $\mathcal{D}_2$.
   - $\mathcal{D}_2'' := \mathcal{D}_2' \setminus \{\boldsymbol{X} \in V_1^\perp : \|\boldsymbol{X} - \mathsf{Med}(\mathcal{D}_2')\|_2 > C\sqrt{p \log n}\}$.
   - Compute the SVD of the $\mathsf{Cov}(\mathcal{D}_2'')$.
   - Let V_2 be the subspace of $p/4$ smallest eigenvectors.
   - Set $\Pi_{V_2}(\widehat{\boldsymbol{\mu}}^{\mathrm{Sp}}) := \mathsf{Mean}(\Pi_{V_2}\mathcal{D}_2'')$ and $\mathcal{D}_3' := \Pi_{(V_1 \oplus V_2)^\perp}\mathcal{D}_2$.
3. ... so on...
4. Define $\widehat{\boldsymbol{\mu}}_n^{\mathrm{Sp}}$ on $V_k = (V_1 \oplus \ldots \oplus V_{k-1})^\perp$ using $\mathcal{D}_k$.
   - Compute the median of $\{\Pi_{V_k}\boldsymbol{X} : \boldsymbol{X} \in \mathcal{D}_k\}$

5. Iterative group soft thresholding

Parameter contamination
Some notation

- We observe $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$ in $\mathbb{R}^p$ such that
$$\boldsymbol{X}_i = \boldsymbol{\mu}^* + \boldsymbol{\theta}_i^* + \boldsymbol{\xi}_i, \quad \boldsymbol{\xi}_i \overset{\text{iid}}{\sim} \mathcal{N}_p(\mathbf{0}, \mathbf{I}_p).$$
- **Goal:** estimate the vector $\boldsymbol{\mu}^*$.
- **Sparsity assumption:** most vectors $\boldsymbol{\theta}_i^*$ are equal to zero.
- $S = \{i : \|\boldsymbol{\theta}_i^*\|_2 > 0\}$ is considered as the set of outliers.
- Vectors $\boldsymbol{\theta}_i$ are unknown nuisance parameters.
- **Matrix notation**: $\mathbf{X} = \boldsymbol{\mu}^* \mathbf{1}_n^\top + \boldsymbol{\Theta}^* + \boldsymbol{\xi}$.
- Auxiliary problem: estimate $L_n(\boldsymbol{\Theta}^*) = \frac{1}{n}\sum_{i=1}^n \boldsymbol{\theta}_i^*$.

Naive idea: Group-lasso estimator

- Group lasso (Chesneau and Hebiri, 2008; Lin and Zhang, 2006; Lounici, Pontil, van de Geer, and Tsybakov, 2011; Meier, van de Geer, and Bühlmann, 2009; Yuan and Lin, 2006):

$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Theta}}) \in \arg\min_{\boldsymbol{\mu},\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^{n} \|\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^{n} \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}.$$

- The above optimization problem is convex and can be solved efficiently even when p and n are large. $\widehat{\boldsymbol{\mu}}$ is exactly the Huber M-estimator (Donoho and Montanari, 2016).

Theorem

If $s \leq n/32$ and $\lambda_i = 6\sqrt{p}$, then, with prob. $\geq 1-\delta$,

$$\|\boldsymbol{L}_n(\widehat{\boldsymbol{\Theta}}) - \boldsymbol{L}_n(\boldsymbol{\Theta}^*)\|_2^2 \lesssim \varepsilon^2 p, \quad \|\widehat{\boldsymbol{\mu}} - \boldsymbol{\mu}^*\|_2^2 \lesssim \frac{p}{n} + \varepsilon^2 p + \frac{\log(2/\delta)}{n}.$$

Idea behind iterative group soft thresholding

- Group lasso:

$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Theta}}) \in \arg\min_{\boldsymbol{\mu},\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^{n} \|\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^{n} \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}.$$

Idea behind iterative group soft thresholding

- Group lasso:

$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Theta}}) \in \arg\min_{\boldsymbol{\mu},\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^{n} \|\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^{n} \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}.$$

- We have

$$\widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \Big\{ \sum_{i=1}^{n} \|\boldsymbol{X}_i - \boldsymbol{\mu} - \widehat{\boldsymbol{\Theta}}_i\|_2^2 \Big\} = L_n(\mathbf{X}) - L_n(\widehat{\boldsymbol{\Theta}}).$$

Idea behind iterative group soft thresholding

- Group lasso:

$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Theta}}) \in \arg\min_{\boldsymbol{\mu},\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^n \|\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^n \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}.$$

- We have

$$\widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \Big\{ \sum_{i=1}^n \|\boldsymbol{X}_i - \boldsymbol{\mu} - \widehat{\boldsymbol{\Theta}}_i\|_2^2 \Big\} = L_n(\mathbf{X}) - L_n(\widehat{\boldsymbol{\Theta}}).$$

- If we set $\boldsymbol{Z}_i = \boldsymbol{X}_i - \big\{L_n(\mathbf{X}) - L_n(\widehat{\boldsymbol{\Theta}})\big\}$, we get

$$\widehat{\boldsymbol{\Theta}} \in \arg\min_{\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^n \|\boldsymbol{Z}_i - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^n \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}$$

  which is the group-soft-thresholding (GST) estimator applied to $\mathbf{Z}$.

Idea behind iterative group soft thresholding

- Group lasso:
$$(\widehat{\boldsymbol{\mu}}, \widehat{\boldsymbol{\Theta}}) \in \arg\min_{\boldsymbol{\mu},\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^n \|\boldsymbol{X}_i - \boldsymbol{\mu} - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^n \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}.$$
- We have
$$\widehat{\boldsymbol{\mu}} = \arg\min_{\boldsymbol{\mu}} \Big\{ \sum_{i=1}^n \|\boldsymbol{X}_i - \boldsymbol{\mu} - \widehat{\boldsymbol{\Theta}}_i\|_2^2 \Big\} = L_n(\mathbf{X}) - L_n(\widehat{\boldsymbol{\Theta}}).$$
- If we set $\boldsymbol{Z}_i = \boldsymbol{X}_i - \big\{L_n(\mathbf{X}) - L_n(\widehat{\boldsymbol{\Theta}})\big\}$, we get
$$\widehat{\boldsymbol{\Theta}} \in \arg\min_{\boldsymbol{\Theta}} \Big\{ \sum_{i=1}^n \|\boldsymbol{Z}_i - \boldsymbol{\theta}_i\|_2^2 + \sum_{i=1}^n \lambda_i \|\boldsymbol{\theta}_i\|_2 \Big\}$$
which is the group-soft-thresholding (GST) estimator applied to $\mathbf{Z}$.
- Some prior work on linear functional estimation suggests to choose
$$\lambda_i = \frac{2p^{1/4}\|\boldsymbol{Z}_i\|_2}{(\|\boldsymbol{Z}_i\|_2^2 - p)_+^{1/2}}.$$
Unfortunately, we can not do that since $\boldsymbol{Z}$ depends on $\widehat{\boldsymbol{\Theta}}$.

Algorithm of IGST

- Start with an estimator $\widehat{\Theta}^0$, for instance, group lasso.
- for $k = 1, \ldots, K$, do

  1) $\boldsymbol{Z}_i = \boldsymbol{X}_i - \{L_n(\mathbf{X}) - L_n(\widehat{\Theta}^{k-1})\}$,

  2) $\lambda_i = \dfrac{2p^{1/4}\|\boldsymbol{Z}_i\|_2}{(\|\boldsymbol{Z}_i\|_2^2 - p)_+^{1/2}}$,

  3) $\widehat{\Theta}_i^k = \text{GST}(\boldsymbol{Z}_i, \lambda_i)$.

- Final estimator:

$$\widehat{\boldsymbol{\mu}}^{\text{IGST}} = L_n(\mathbf{X}) - L_n(\widehat{\Theta}^K).$$

Iterative group soft thresholding
Risk bound

Theorem 5 (Collier and Dalalyan, 2017)

Let $\nu > 0$ and assume the IGST estimator is run for $K = \log_2(1/\nu) + \log\log p$ iterations.
There are constants $c, C > 0$ such that for every $\varepsilon \leq c$, in an event of probability $\geq 1 - e^{-p/8}$, the IGST estimator satisfies

$$\|\widehat{\boldsymbol{\mu}}^{\mathrm{IGST}} - \boldsymbol{\mu}\|_2^2 \leq C\left(\frac{p}{n} + \varepsilon^2 + \left(p\varepsilon^4\right)^{1-\nu}\right).$$

Iterative group soft thresholding

Risk bound

Theorem 5 (Collier and Dalalyan, 2017)

Let $\nu > 0$ and assume the IGST estimator is run for $K = \log_2(1/\nu) + \log\log p$ iterations.
There are constants $c, C > 0$ such that for every $\varepsilon \le c$, in an event of probability $\ge 1 - e^{-p/8}$, the IGST estimator satisfies

$$\|\widehat{\boldsymbol{\mu}}^{\mathrm{IGST}} - \boldsymbol{\mu}\|_2^2 \le C\bigg(\frac{p}{n} + \varepsilon^2 + \Big(p\varepsilon^4\Big)^{1-\nu}\bigg).$$

- Valid for $\mathcal{M}_n^{\mathrm{PC}}(p, \varepsilon, \boldsymbol{\mu}^*)$.

Iterative group soft thresholding

Risk bound

Theorem 5 (Collier and Dalalyan, 2017)

Let $\nu > 0$ and assume the IGST estimator is run for $K = \log_2(1/\nu) + \log\log p$ iterations.
There are constants $c, C > 0$ such that for every $\varepsilon \le c$, in an event of probability $\ge 1 - e^{-p/8}$, the IGST estimator satisfies

$$\|\widehat{\boldsymbol{\mu}}^{\rm IGST} - \boldsymbol{\mu}\|_2^2 \le C\bigg(\frac{p}{n} + \varepsilon^2 + \Big(p\varepsilon^4\Big)^{1-\nu}\bigg).$$

- Valid for $\mathcal{M}_n^{\rm PC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity $O(np\log\log p)$

Iterative group soft thresholding

Risk bound

Theorem 5 (Collier and Dalalyan, 2017)

Let $\nu > 0$ and assume the IGST estimator is run for $K = \log_2(1/\nu) + \log\log p$ iterations.
There are constants $c, C > 0$ such that for every $\varepsilon \leq c$, in an event of probability $\geq 1 - e^{-p/8}$, the IGST estimator satisfies

$$\|\widehat{\boldsymbol{\mu}}^{\rm IGST} - \boldsymbol{\mu}\|_2^2 \leq C\bigg(\frac{p}{n} + \varepsilon^2 + \Big(p\varepsilon^4\Big)^{1-\nu}\bigg).$$

- Valid for $\mathcal{M}_n^{\rm PC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity $O(np\log\log p)$
- Exponential in p decay of probability.

Iterative group soft thresholding
Risk bound

Theorem 5 (Collier and Dalalyan, 2017)

Let $\nu > 0$ and assume the IGST estimator is run for $K = \log_2(1/\nu) + \log\log p$ iterations.
There are constants $c, C > 0$ such that for every $\varepsilon \le c$, in an event of probability $\ge 1 - e^{-p/8}$, the IGST estimator satisfies

$$\|\widehat{\boldsymbol{\mu}}^{\rm IGST} - \boldsymbol{\mu}\|_2^2 \le C\bigg(\frac{p}{n} + \varepsilon^2 + \Big(p\varepsilon^4\Big)^{1-\nu}\bigg).$$

- Valid for $\mathcal{M}_n^{\rm PC}(p, \varepsilon, \boldsymbol{\mu}^*)$.
- Overall complexity $O(np\log\log p)$
- Exponential in p decay of probability.
- The rate is optimal in the regime $\varepsilon = O(p^{-1/2} \vee n^{-1/4})$.

Summary

- The ellipsoid method for robust estimation:
  - achieves the minimax rate on $\mathcal{M}_n^{\rm AC}(p,\varepsilon,\boldsymbol{\mu}^*)$ up to an extra factor $\log(1/\varepsilon)$.
  - Complexity of 1 iteration $O(np^2)$.
  - Exponential in p decay of probability.
  - Poly number of iterations ?
- The spectral method for robust estimation:
  - achieves the minimax rate on $\mathcal{M}_n^{\rm AC}(p,\varepsilon,\boldsymbol{\mu}^*)$ up to an extra factor $(\log p)^2\log n$.
  - Overall complexity $O(np^2)$.
  - Polynomial in p decay of probability.
- The iterative group-soft-thresholding:
  - achieves the minimax rate on $\mathcal{M}_n^{\rm PC}(p,\varepsilon,\boldsymbol{\mu}^*)$ without any extra factor when $\varepsilon = O(p^{-1/2}\vee n^{-1/4})$.
  - Overall complexity $O(np)$.
  - Exponential in p decay of probability.


thank you
danke
謝謝
ngiyabonga
teşekkür ederim
gracias
merci
obrigado
dziękuję
спасибо
bedankt
go raibh maith agat
mochchakkeram
sukriya
kop khun krap
terima kasih
grazie
arigatô
dakujem
takk
tapadh leat
dank je
мерси
감사합니다
ευχαριστώ
sagolun
hvala
mauruuru
taiku
Баярлалаа
рахмат
mersi
barka
welalin
tack
spas
misaotra
matondo
paldies
grazzi
mahalo
хвала
asante
manana
obrigada
tenki
murakoze
chokrane
mamnun
дякую
trugarez
shukriya
mercê
dhanyavadagalu
diolch
xièxie
rahmet
tanemirt
najis tuke
ありがとう
kam sah hamnida
rahmat
didi madloba
mèsi
sobodi
dekuji
enkosi
bayarlalaa
gracie
köszönöm
kiitos
dankie
dhanyavad
nandri
nanni
faafetai lava
vinaka
спасибі
blagodaram
kia ora
akun dankon
aciu
chnorakaloutioun
gratias ago
gràcies
sulpáy
djiere dieuf
tau

References I

Christophe Chesneau and Mohamed Hebiri. Some theoretical results on the grouped variables Lasso. Math. Methods Statist., 17(4):317–326, 2008.

Olivier Collier and Arnak S. Dalalyan. Minimax estimation of a multidimensional linear functional in sparse gaussian models and robust estimation of the mean. submitted 1712.05495, arXiv, December 2017. URL https://arxiv.org/abs/1712.05495.

Ilias Diakonikolas, Gautam Kamath, Daniel M. Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high dimensions without the computational intractability. In IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, USA, pages 655–664, 2016.

David Donoho and Andrea Montanari. High dimensional robust m-estimation: asymptotic variance via approximate message passing. Probability Theory and Related Fields, 166(3):935–969, Dec 2016.

K. A. Lai, A. B. Rao, and S. Vempala. Agnostic estimation of mean and covariance. In 2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS), pages 665–674, Oct 2016.

References II

Yi Lin and Hao Helen Zhang. Component selection and smoothing in multivariate nonparametric regression. Ann. Statist., 34(5):2272–2297, 2006.

Karim Lounici, Massimiliano Pontil, Sara van de Geer, and Alexandre B. Tsybakov. Oracle inequalities and optimal inference under group sparsity. Ann. Statist., 39(4):2164–2204, 2011.

Lukas Meier, Sara van de Geer, and Peter Bühlmann. High-dimensional additive modeling. Ann. Statist., 37(6B):3779–3821, 2009.

Ming Yuan and Yi Lin. Model selection and estimation in regression with grouped variables. J. R. Stat. Soc. Ser. B Stat. Methodol., 68(1):49–67, 2006.